M.Sc. IT-Systems Engineering

Pflichtmodule (ITSE)

Analyse (HPI-ITSE-A)

6 **Graphenalgorithmen**
Vorlesung/Übung/ 4 — *Friedrich, Tobias*, *Skretas, Georgios*

4 **Big Data Systeme**
Vorlesung/4 — *Rabl, Tilmann*, *Boissier, Martin*, *Salazar Diaz, Ricardo*, *Strassenburg, Nils*

Entwurf (HPI-ITSE-E)

018 **Kryptographie**
Vorlesung/Übung/ 4 — *Lehmann, Anja*, *Dayanikli, Dennis Kenan*

Die Vorlesung gibt eine umfassende Einführung in die moderne Kryptographie und die Grundkonzepte der beweisbaren Sicherheit. Dazu werden formale Angreifermodelle definiert und die Sicherheit der vorgestellten Kryptoverfahren unter wohldefinierten Komplexitätsannahmen in diesem Angreifermodell nachgewiesen. Der Vorlesung dient auch als Grundlage für andere Kurse zur Kryptographie, die vom Lehrstuhl angeboten werden.

Content of teaching

- Informationstheoretische vs. Komplexitätstheoretische Sicherheit
- Symmetrische Kryptographie
  - Symmetrische Verschlüsselung
  - Pseudozufallsfunktionen
  - Message Authentication Codes (MAC)
  - Hash-Funktionen
  - Authenticated Encryption
- Asymmetrische Kryptographie
  - Diffie-Hellman Schlüsselaustausch
  - Public-Key Verschlüsselung
  - Digitale Signaturen

Die Vorlesung setzt Grundkenntnisse in Mathematik und theoretischer Informatik voraus, insbesondere müssen die formale mathematische Sprache und elementare Beweistechniken (Widerspruchsbeweis) problemlos angewandt werden können. Wenn diese Kenntnisse nicht vorhanden sind, wird empfohlen dieses Wissen vor der Vorlesung selbstständig zu erwerben, z.B. durch die Teilnahme an den Vorlesungen Mathematik I oder II (ITSE-Bachelor). In den ersten Vorlesungswochen wird es voraussichtlich auch zusätzliche Übungstermine und -materialien geben, in denen elementare Grundlagen aufgefrischt werden können.

6 **Graphenalgorithmen**
Vorlesung/Übung/ 4 — *Friedrich, Tobias*, *Skretas, Georgios*

Konstruktion (HPI-ITSE-K)

018 **Kryptographie**

Vorlesung/Übung/ 4

Die Vorlesung gibt eine umfassende Einführung in die moderne Kryptographie und die Grundkonzepte der beweisbaren Sicherheit. Dazu werden formale Angreifermodelle definiert und die Sicherheit der vorgestellten Kryptoverfahren unter wohldefinierten Komplexitätsannahmen in diesem Angreifermodell nachgewiesen. Der Vorlesung dient auch als Grundlage für andere Kurse zur Kryptographie, die vom Lehrstuhl angeboten werden.

Content of teaching

- Informationstheoretische vs. Komplexitätstheoretische Sicherheit
- Symmetrische Kryptographie
  - Symmetrische Verschlüsselung
  - Pseudozufallsfunktionen
  - Message Authentication Codes (MAC)
  - Hash-Funktionen
  - Authenticated Encryption
- Asymmetrische Kryptographie
  - Diffie-Hellman Schlüsselaustausch
  - Public-Key Verschlüsselung
  - Digitale Signaturen

Die Vorlesung setzt Grundkenntnisse in Mathematik und theoretischer Informatik voraus, insbesondere müssen die formale mathematische Sprache und elementare Beweistechniken (Widerspruchsbeweis) problemlos angewandt werden können. Wenn diese Kenntnisse nicht vorhanden sind, wird empfohlen dieses Wissen vor der Vorlesung selbstständig zu erwerben, z.B. durch die Teilnahme an den Vorlesungen Mathematik I oder II (ITSE-Bachelor). In den ersten Vorlesungswochen wird es voraussichtlich auch zusätzliche Übungstermine und -materialien geben, in denen elementare Grundlagen aufgefrischt werden können.

Lehmann, Anja
Dayanikli, Dennis Kenan

035 **Advanced Topics in Software Engineering: Automation and AI**

Vorlesung/4

Giese, Holger
Barkowsky, Matthias
Adriano, Christian
Ghahremani, Sona

In software engineering, like many other engineering disciplines, we on the one hand want to build solutions with the best possible quality while we on the other hand must adhere to predetermined budget and time constraints. Furthermore, often not enough qualified software engineers are available. Therefore, improving the productivity during software development and operation as well as the quality of the outcomes by automating activities partially or completely using software itself has been a major area for innovations since the early days of programming and software engineering.
Nowadays, many software engineering activities benefit from a high degree of automation and very often we take that automation for granted and are hardly aware of it anymore. Also, often the considered software systems have become so complex that they can only be developed, operated, and evolved by using largely automated approaches for various software engineering activities.
Automation in software engineering has the goal to partially or fully execute software engineering activities with minimal human intervention, thereby significantly increasing both quality and productivity. Automation successfully encompasses a broad range of activities, for instance, requirements definition, specification, software architecture, software design and synthesis, implementation, modeling, testing, quality assurance, verification, validation, maintenance, evolution, configuration management, deployment, reengineering, reuse and visualization. (…)
Also artificial intelligence is nowadays used to enhance existing software systems or make new beforehand not feasible software systems possible. Therefore, software engineering activities and outcomes have to be adjusted so that software solutions can benefit from integrated features realized with artificial intelligence. This requires that a clear strategy on how to use artificial intelligence in a software is established and that all aspects of software development and operation are appropriately adjusted to ensure that the employed combination of traditional software and artificial intelligence results in the required quality.
Therefore, we will look in this course at first into the advanced development of systems using automation for software engineering including artificial Intelligence as well as secondly into software engineering for the development of advanced systems that employ artificial intelligence. Furthermore, we will also investigate the operation of systems and how automation and in particular artificial intelligence can help there. Finally, we will discuss the case where automation and in particular artificial intelligence is used for development and operation and employed for the system itself at the same time.
We will in addition to the discussions in the lecture explore the key challenges also with small projects in the exercises and will collect at the beginning of the course suggestions for artificial intelligence tools to consider for the small projects or student presentations.
1. https://www.infoworld.com/article/3489925/github-survey-finds-nearly-all-developers-using-ai-coding-tools.html
2. https://research.google/blog/ai-in-software-engineering-at-google-progress-and-the-path-ahead/

Exam:
The grading process takes into account two components:
The results of the hands-on projects accompanying the lecture, with each project graded individually.
A final exam at the end of the semester. Depending on the number of course participants, the exam will either be oral or written.
Students will be required to pass both graded components. In particular, completing all hands-on projects to an adequate level is required for admission to the exam.
The final grade will either be composed of the average project grade (50%) and the exam grade (50%) OR the exam grade (100%) only, depending on which grading scheme yields a better result for each student individually.

Maintenance (HPI-ITSE-M)

035 **Advanced Topics in Software Engineering: Automation and AI**

Vorlesung/4

Giese, Holger
Barkowsky, Matthias
Adriano, Christian
Ghahremani, Sona

In software engineering, like many other engineering disciplines, we on the one hand want to build solutions with the best possible quality while we on the other hand must adhere to predetermined budget and time constraints. Furthermore, often not enough qualified software engineers are available. Therefore, improving the productivity during software development and operation as well as the quality of the outcomes by automating activities partially or completely using software itself has been a major area for innovations since the early days of programming and software engineering.
Nowadays, many software engineering activities benefit from a high degree of automation and very often we take that automation for granted and are hardly aware of it anymore. Also, often the considered software systems have become so complex that they can only be developed, operated, and evolved by using largely automated approaches for various software engineering activities.
Automation in software engineering has the goal to partially or fully execute software engineering activities with minimal human intervention, thereby significantly increasing both quality and productivity. Automation successfully encompasses a broad range of activities, for instance, requirements definition, specification, software architecture, software design and synthesis, implementation, modeling, testing, quality assurance, verification, validation, maintenance, evolution, configuration management, deployment, reengineering, reuse and visualization. (…)
Also artificial intelligence is nowadays used to enhance existing software systems or make new beforehand not feasible software systems possible. Therefore, software engineering activities and outcomes have to be adjusted so that software solutions can benefit from integrated features realized with artificial intelligence. This requires that a clear strategy on how to use artificial intelligence in a software is established and that all aspects of software development and operation are appropriately adjusted to ensure that the employed combination of traditional software and artificial intelligence results in the required quality.
Therefore, we will look in this course at first into the advanced development of systems using automation for software engineering including artificial Intelligence as well as secondly into software engineering for the development of advanced systems that employ artificial intelligence. Furthermore, we will also investigate the operation of systems and how automation and in particular artificial intelligence can help there. Finally, we will discuss the case where automation and in particular artificial intelligence is used for development and operation and employed for the system itself at the same time.
We will in addition to the discussions in the lecture explore the key challenges also with small projects in the exercises and will collect at the beginning of the course suggestions for artificial intelligence tools to consider for the small projects or student presentations.
1. https://www.infoworld.com/article/3489925/github-survey-finds-nearly-all-developers-using-ai-coding-tools.html
2. https://research.google/blog/ai-in-software-engineering-at-google-progress-and-the-path-ahead/

Exam:
The grading process takes into account two components:
The results of the hands-on projects accompanying the lecture, with each project graded individually.
A final exam at the end of the semester. Depending on the number of course participants, the exam will either be oral or written.
Students will be required to pass both graded components. In particular, completing all hands-on projects to an adequate level is required for admission to the exam.
The final grade will either be composed of the average project grade (50%) and the exam grade (50%) OR the exam grade (100%) only, depending on which grading scheme yields a better result for each student individually.

4 **Big Data Systeme**

Vorlesung/4

Rabl, Tilmann
Boissier, Martin
Salazar Diaz, Ricardo
Strassenburg, Nils

Management-Kompetenzen (HPI-MK)

040 **Führungskompetenz - über die harten Auswirkungen der Soft Skills**

Blockseminar/2

Heidemann, Michael Karl
Unger, Eugen
Fuerstenberg, Anja

Fachliche Kompetenzen werden in Unternehmen als selbstverständlich vorausgesetzt. Das Seminar geht von der These aus, dass mit jedem Karriereschritt in der Hierarchie auch die Anforderungen an soziale Kompetenz (Kommunikationsfähigkeit, Konfliktfähigkeit, Werteorientierung) steigen.
Modul 1 - Referent Michael Karl Heidemann
Führung in Veränderungsprozessen: Unternehmenskultur gestalten
Verantwortung in Unternehmen zu tragen, heißt heute vor allem, Veränderungsprozesse zu initiieren, zu begleiten und erfolgreich zu machen. Welche Herausforderung bedeutet das für Führungskräfte? Wodurch ist die Unternehmenskultur eines Unternehmens bestimmt? Welche Faktoren spielen grundsätzlich eine Rolle, welche sind im Alltag wirksam? Lässt sich die Führungskultur eines Unternehmens beeinflussen und wenn ja – wie? Im ersten Modul der Reihe wird eine grundsätzliche, an der Führungsverantwortung orientierte Sicht auf das Thema entfaltet.

- Was ist Unternehmenskultur?
- Welche Bedeutung hat sie für den Erfolg des Unternehmens?
- Kann man Menschen verändern?
- Kann man Unternehmen verändern?
- Kulturelle Aspekte im Change Management
- Führung als Identitätsstiftung
- Herausforderungen in Veränderungsprozessen
- Autonomie und Heteronomie im Führungsalltag

Modul 2 - Referent Eugen Unger
Führungsalltag: Führungssituationen und Führungskommunikation
Führung beruht, wie alles soziale Handeln, auf Verhaltensmustern, die weitgehend automatisch, also unbewußt ablaufen. Das eigene Handeln an selbst entwickelten Qualitätsmaßstäben zu orientieren, bedeutet demnach Bewusstsein zu schaffen. Die Teilnehmer reflektieren ihr Führungsverständnis, indem sie sich mit ihren eigenen Annahmen und daraus resultierenden Verhaltensstrategien auseinandersetzen. Auf diese Weise bietet das Format einen diskursiven Rahmen für relevante Führungsthemen des Alltags und fördert damit ein klares Rollenverständnis als Führende.

- Selbstverständnis als Führungskraft
- Rollenanforderungen zwischen Zielen und Bedürfnissen
- Anerkennung, Kritik und Potentialentwicklung
- Führungskommunikation bewußt gestalten
- Feedbacksicherheit
- Motivation und Demotivatoren
- Zusammenspiel der Führungsinstrumente

Exam: Die Leistungserfassung erfolgt im Rahmen einer mündlichen Prüfung (Kolloquium).

041 **Intrapersonelle & Interpersonelle Kompetenzen**

Blockseminar/2

Leidenfrost, Jana
Fuerstenberg, Anja

106 **Management Essentials**

Blockseminar/2

The students learn about the most important aspects of managing organizations and of managing people in organizations and how to apply this knowledge to concrete challenges.

This course offers an overview of the main topics of management. We will first cover the basics of management *of* organizations (strategic leadership) and will then turn to management *in* organizations (people management). With regard to the latter, the topics include leadership and motivation, employee satisfaction, personnel selection, training and development, and employee evaluation and compensation. Management knowledge is essential for all those who at some point wish to start their own companies or strive to occupy leadership positions in organizations.

Conveyed competencies:

Knowledge-related competencies: strategic management; methods in management research; personnel selection; job and work design; training and development; motivation; satisfaction; leadership; personnel evaluation; personnel compensation.
Methodological competencies; case study analysis; presentation techniques.
Social competencies; group work and discussions.

Exam: The grade will be calculated on the basis of a group presentation (30%) and a written assignment (70%). Both the group presentation and the written assignment will focus on management aspects in organizations that the students select themselves. Further details will be provided at the beginning of the course.

Kearney, Eric
Fuerstenberg, Anja

049 **Managing stakeholders – The psychology and neuroscience of successfully influencing others**

Blockseminar/2

Frank, Franziska
Fuerstenberg, Anja

050 **Power and Power Misuse in Organizations**

Blockseminar/2

Drath, Karsten
Fuerstenberg, Anja

Business Process and Enterprise Technologies (BPET)

Konzepte und Methoden (HPI-BPET-K)

027 **Process Mining**

Vorlesung/Übung/ 2

Part 1: Power in Organizations. What is it? (0.75 days)
Part 2: Destructive Leaders – Born or made? (0.75 days)
Part 3: Power Misuse in Organizations (0.75 days)
Part 4: Managing Power in Organizations (0.75 day)

Exam:

Class presentation (50%)
Written exam (50%)

Leopold, Henrik
Weske, Mathias

5 **Global Team-Based Innovation I**

Projektseminar/4

Uebernickel, Falk
Beermann, Vincent
Enkmann, Jan
Rolfes, Theresa Maria
Cauderay, Virginie
Wuttke, Tobias

Global Team-based Innovation (GTI) is a course designated for master students of the Hasso Plattner Institute (HPI) and the University of Potsdam (UP).
In our course, students apply IT knowledge to engineer digital solutions for real business challenges provided by prominent global companies. We follow the Design Thinking methodology to innovate on wicked problems given by our project partners. Within GTI, HPI students collaborate with students from other leading global universities: HPI is a partner in ME310 (for projects with the Stanford University) as well as part of the SUGAR Network for Design Innovation (for projects with other global universities).
https://hpi.de/uebernickel/teaching/global-team-based-innovation-gti-design-thinking.html
This class is exclusively available to students who have been accepted through our application process.

Exam
- Project work (20%)
  - Individual participation during lectures, group meetings and in project work
  - Stakeholder management
  - Project management (sticking to deadlines, etc.)
- Milestone presentations (20%)
  - GTI 1: Fall & winter presentation
  - GTI 2: Final presentation
- Tangible outcomes (20%)
  - One-Pagers for corporate partners
  - Intermediate prototypes
- Milestone documentations (40%)
  - GTI 1: Fall & winter documentation
  - GTI 2: Final documentation & videos

The estimated workload is 2-3 days per week.

Goals:
Students from Potsdam and leading global partner universities tackle design innovation challenges posed by global corporations. The 9 months (2 semesters) course focuses on the application of IT knowledge for engineering solutions to real business challenges. Further, we put emphasis on teaching students human-centered innovation methods and processes required for designers, engineers, and project managers of the future.
Within the projects, students go through an intense and iterative process of need finding, ideation, and rapid prototyping to create and evaluate new concepts. Company involvement provides the reality check necessary for teams to improve their innovation abilities. The team is supported by a professional coach, corporate liaisons, and faculty advisors.
Projects typically involve systems integration and include a mix of mechanical, electronic and software design. The results of all projects are real prototypes that have a user-centric design, are economically viable and technically feasible.

002 **Digital Health and Research Systems, Data Interoperability**

Vorlesung/Seminar/4

Heitmann, Kai U.
Thun, Sylvia
Prasser, Fabian
Arnrich, Bert

Technologien und Werkzeuge (HPI-BPET-T)

027 **Process Mining**

Vorlesung/Übung/2

Leopold, Henrik
Weske, Mathias

5 **Global Team-Based Innovation I**

Projektseminar/4

Uebernickel, Falk
Beermann, Vincent
Enkmann, Jan
Rolfes, Theresa Maria
Cauderay, Virginie
Wuttke, Tobias

Global Team-based Innovation (GTI) is a course designated for master students of the Hasso Plattner Institute (HPI) and the University of Potsdam (UP).
In our course, students apply IT knowledge to engineer digital solutions for real business challenges provided by prominent global companies. We follow the Design Thinking methodology to innovate on wicked problems given by our project partners. Within GTI, HPI students collaborate with students from other leading global universities: HPI is a partner in ME310 (for projects with the Stanford University) as well as part of the SUGAR Network for Design Innovation (for projects with other global universities).
https://hpi.de/uebernickel/teaching/global-team-based-innovation-gti-design-thinking.html
This class is exclusively available to students who have been accepted through our application process.

Exam
- Project work (20%)
  - Individual participation during lectures, group meetings and in project work
  - Stakeholder management
  - Project management (sticking to deadlines, etc.)
- Milestone presentations (20%)
  - GTI 1: Fall & winter presentation
  - GTI 2: Final presentation
- Tangible outcomes (20%)
  - One-Pagers for corporate partners
  - Intermediate prototypes
- Milestone documentations (40%)
  - GTI 1: Fall & winter documentation
  - GTI 2: Final documentation & videos

The estimated workload is 2-3 days per week.

Goals:
Students from Potsdam and leading global partner universities tackle design innovation challenges posed by global corporations. The 9 months (2 semesters) course focuses on the application of IT knowledge for engineering solutions to real business challenges. Further, we put emphasis on teaching students human-centered innovation methods and processes required for designers, engineers, and project managers of the future.
Within the projects, students go through an intense and iterative process of need finding, ideation, and rapid prototyping to create and evaluate new concepts. Company involvement provides the reality check necessary for teams to improve their innovation abilities. The team is supported by a professional coach, corporate liaisons, and faculty advisors.
Projects typically involve systems integration and include a mix of mechanical, electronic and software design. The results of all projects are real prototypes that have a user-centric design, are economically viable and technically feasible.

002 **Digital Health and Research Systems, Data Interoperability**

Vorlesung/Seminar/4

Heitmann, Kai U.
Thun, Sylvia
Prasser, Fabian
Arnrich, Bert

Spezialisierung (HPI-BPET-S)

027 **Process Mining**

Vorlesung/Übung/2

Leopold, Henrik
Weske, Mathias

5	**Global Team-Based Innovation I**		
	Projektseminar/4	Global Team-based Innovation (GTI) is a course designated for master students of the Hasso Plattner Institute (HPI) and the University of Potsdam (UP). In our course, students apply IT knowledge to engineer digital solutions for real business challenges provided by prominent global companies. We follow the Design Thinking methodology to innovate on wicked problems given by our project partners. Within GTI, HPI students collaborate with students from other leading global universities: HPI is a partner in ME310 (for projects with the Stanford University) as well as part of the SUGAR Network for Design Innovation (for projects with other global universities). https://hpi.de/uebernickel/teaching/global-team-based-innovation-gti-design-thinking.html This class is exclusively available to students who have been accepted through our application process. Exam Project work (20%) Individual participation during lectures, group meetings and in project work Stakeholder management Project management (sticking to deadlines, etc.) Milestone presentations (20%) GTI 1: Fall & winter presentation GTI 2: Final presentation Tangible outcomes (20%) One-Pagers for corporate partners Intermediate prototypes Milestone documentations (40%) GTI 1: Fall & winter documentation GTI 2: Final documentation & videos The estimated workload is 2-3 days per week. Goals: Students from Potsdam and leading global partner universities tackle design innovation challenges posed by global corporations. The 9 months (2 semesters) course focuses on the application of IT knowledge for engineering solutions to real business challenges. Further, we put emphasis on teaching students human-centered innovation methods and processes required for designers, engineers, and project managers of the future. Within the projects, students go through an intense and iterative process of need finding, ideation, and rapid prototyping to create and evaluate new concepts. Company involvement provides the reality check necessary for teams to improve their innovation abilities. The team is supported by a professional coach, corporate liaisons, and faculty advisors. Projects typically involve systems integration and include a mix of mechanical, electronic and software design. The results of all projects are real prototypes that have a user-centric design, are economically viable and technically feasible.	*Uebernickel, Falk* *Beermann, Vincent* *Enkmann, Jan* *Rolfes, Theresa Maria* *Cauderay, Virginie* *Wuttke, Tobias*

Human Computer Interaction and Computer Graphics Technology (HCGT)

Konzepte und Methoden (HPI-HCGT-K)

9	**HCI Project Seminar on Virtual Reality and Personal Fabrication**		
	Seminar/Praktikum /4		*Baudisch, Patrick*
3	**Creating Interactive 3D Web Apps with TypeScript**		
	Projektseminar/4		*Baudisch, Patrick*
4	**Algorithmic folding**		
	Vorlesung/4		*Baudisch, Patrick* *Abdullah, Muhammad* *Rambold, Lukas*
0	**Explaining and Visualizing AI**		
	Seminar/Praktikum /4		*Burmeister, Josafat-Mattias* *Cech, Tim* *Doellner, Juergen*

2	**Spatial Data: Processing and Visualization Techniques**	
	Seminar/Praktikum /4	*Richter, Rico* *Wegen, Ole* *Hildebrand, Justus* *Schulz, Sebastian* *Burmeister, Josafat-Mattias*

Techniken und Werkzeuge (HPI-HCGT-T)

9	**HCI Project Seminar on Virtual Reality and Personal Fabrication**	
	Seminar/Praktikum /4	*Baudisch, Patrick*
3	**Creating Interactive 3D Web Apps with TypeScript**	
	Projektseminar/4	*Baudisch, Patrick*
4	**Algorithmic folding**	
	Vorlesung/4	*Baudisch, Patrick* *Abdullah, Muhammad* *Rambold, Lukas*
0	**Explaining and Visualizing AI**	
	Seminar/Praktikum /4	*Burmeister, Josafat-Mattias* *Cech, Tim* *Doellner, Juergen*
2	**Spatial Data: Processing and Visualization Techniques**	
	Seminar/Praktikum /4	*Richter, Rico* *Wegen, Ole* *Hildebrand, Justus* *Schulz, Sebastian* *Burmeister, Josafat-Mattias*

Spezialisierung (HPI-HCGT-S)

9	**HCI Project Seminar on Virtual Reality and Personal Fabrication**	
	Seminar/Praktikum /4	*Baudisch, Patrick*
3	**Creating Interactive 3D Web Apps with TypeScript**	
	Projektseminar/4	*Baudisch, Patrick*
4	**Algorithmic folding**	
	Vorlesung/4	*Baudisch, Patrick* *Abdullah, Muhammad* *Rambold, Lukas*
0	**Explaining and Visualizing AI**	
	Seminar/Praktikum /4	*Burmeister, Josafat-Mattias* *Cech, Tim* *Doellner, Juergen*
2	**Spatial Data: Processing and Visualization Techniques**	
	Seminar/Praktikum /4	*Richter, Rico* *Wegen, Ole* *Hildebrand, Justus* *Schulz, Sebastian* *Burmeister, Josafat-Mattias*

Internet, Security and Algorithm Engineering (ISAE)

Konzepte und Methoden (HPI-ISAE-K)

3	**Creating Interactive 3D Web Apps with TypeScript**	
	Projektseminar/4	*Baudisch, Patrick*
024	**Large Language Models and Computer Vision Research Seminar**	
	Projektseminar/4	*de Melo, Gerard* *Zhang, Jingyi*
3	**Network Security in Practice**	
	Seminar/Praktikum /4	*Najafi, Peyman* *Cheng, Feng*

025 **Computing on Encrypted Data**

Vorlesung/Übung/ 2

This course offers an introduction to cryptographic techniques that enable computation over encrypted data, with a central focus on Homomorphic Encryption. We will follow a practical and engineering-focused approach: while we will touch on essential theoretical concepts, the primary emphasis will be on equipping participants with the skills needed to implement these techniques in real-world applications. The course will comprise a hands-on project where participants will apply what they've learned to develop a functional cryptographic system.

Exam: The grading will be based on a final exam (70%) and a practical project evaluation (30%). The final exam will be oral, unless too many participant register

Content of teaching:

- Definitions and model
- Early constructions
- Current, lattice-based constructions
- Multiparty homomorphic encryption & Secure multiparty computations
- Implementation

Prerequisites:

- Introduction to cryptography: encryption, security property and game-based proofs.
- Basic discrete mathematics: modular algebra, very basic group and ring theory.
- Programming: current HE implementation are in C++ and Go.

Mouchet, Christian
Lehmann, Anja

0 **Mobile Security**

Vorlesung/Übung/ 4

This lecture covers mobile security on an application and system level, with many hands-on exercises. Students will learn state-of-the-art security concepts for both, iOS and Android, and will be able to perform security testing of mobile apps, mobile malware analysis, as well as testing security-critical components within mobile operating systems.

https://moodle.hpi.de/course/view.php?id=798
This lecture covers mobile security on an application and system level, with many hands-on exercises. Students will learn state-of-the-art security concepts for both iOS and Android. They will be able to perform security testing of mobile apps, mobile malware analysis, and testing security-critical components within mobile operating systems.

Course contents include:

- Threat modeling for mobile devices and apps,
- building mobile applications with Xcode and Android Studio,
- application security and testing,
- mobile malware capabilities and detection,
- operating system internals, such as inter-process communication, threads, ...,
- kernel and firmware security,
- mobile forensics, and
- wireless security.

Grading is based on practical exercises and the final exam.

- Assignments (50%)
- Written exam, 90 minutes (50%)

Classen, Jiska

018 **Kryptographie**

Vorlesung/Übung/ 4

Die Vorlesung gibt eine umfassende Einführung in die moderne Kryptographie und die Grundkonzepte der beweisbaren Sicherheit. Dazu werden formale Angreifermodelle definiert und die Sicherheit der vorgestellten Kryptoverfahren unter wohldefinierten Komplexitätsannahmen in diesem Angreifermodell nachgewiesen. Der Vorlesung dient auch als Grundlage für andere Kurse zur Kryptographie, die vom Lehrstuhl angeboten werden.

Content of teaching

- Informationstheoretische vs. Komplexitätstheoretische Sicherheit
- Symmetrische Kryptographie
  - Symmetrische Verschlüsselung
  - Pseudozufallsfunktionen
  - Message Authentication Codes (MAC)
  - Hash-Funktionen
  - Authenticated Encryption
- Asymmetrische Kryptographie
  - Diffie-Hellman Schlüsselaustausch
  - Public-Key Verschlüsselung
  - Digitale Signaturen

Die Vorlesung setzt Grundkenntnisse in Mathematik und theoretischer Informatik voraus, insbesondere müssen die formale mathematische Sprache und elementare Beweistechniken (Widerspruchsbeweis) problemlos angewandt werden können. Wenn diese Kenntnisse nicht vorhanden sind, wird empfohlen dieses Wissen vor der Vorlesung selbstständig zu erwerben, z.B. durch die Teilnahme an den Vorlesungen Mathematik I oder II (ITSE-Bachelor). In den ersten Vorlesungswochen wird es voraussichtlich auch zusätzliche Übungstermine und -materialien geben, in denen elementare Grundlagen aufgefrischt werden können.

Lehmann, Anja
Dayanikli, Dennis Kenan

017 **Digital Entomology: Tracking and Tackling Cyber Bugs**

Seminar/3

Cybersecurity attacks happen frequently and have severe impact. Bugs in digital systems make these attacks possible. In this seminar, we'll take a look into these bugs, why they happen, how they can be exploited, and what could be done to mitigate them. We're collecting and studying cyber bugs – and you'll all be digital entomologists!
https://moodle.hpi.de/course/edit.php?id=799

The seminar follows a weekly schedule. Each week, we'll talk about recent, impactful bugs. The research talks will be split into bugs presented by the lecturer as well as bugs presented by students. We aim at covering highly diverse and recent bugs and bug classes, such as:

- web and browser security,
- internet-facing services including firewalls, mail, ...,
- binary exploitation,
- real-world bugs in cryptographic implementations,
- hardware bugs,
- ...

Students can pick the bugs they present on their own, but there'll be some moderation to ensure no duplicate bugs and a high variety.

Some experience in the area of cyber security is recommended. You should be able to follow technical writeups about bugs and how they were exploited in order to give presentations about these bugs.

Exam

- 70% Presentations (two 30 minute presentations per student – that means two bugs being presented; each presentation is 20 minutes talk + 10 minutes Q&A)
- 30% Creating quizzes (create multiple choice quizzes for two pressentations)
- Passing all multiple choice quizzes during the semester with at least 75% is mandatory, multiple attempts are allowed.

Classen, Jiska

0 **Cyber Security Management**
Vorlesung/Übung/4 — *Doerr, Christian*

6 **Graphenalgorithmen**
Vorlesung/Übung/4 — *Friedrich, Tobias; Skretas, Georgios*

6 **Advanced Competitive Programming 2**
Vorlesung/4 — *Friedrich, Tobias; Simonov, Kirill; Cohen, Sarel*

6 **Trends in Betriebssystemen und Middleware (Forschungsseminar)**
Seminar/2 — *Polze, Andreas*

1 **Mobilkommunikation**
Vorlesung/Übung/4 — For details, please check Moodle. — *Karl, Holger*

8 **Advanced Machine Learning Seminar**
Seminar/4 — *Lippert, Christoph*

019 **Modern and Secure Internet: Design and Operations**
Vorlesung/4 — *Bajpai, Vaibhav; Ververis, Vasileios*

Technologien und Werkzeuge (HPI-ISAE-T)

3 **Creating Interactive 3D Web Apps with TypeScript**
Projektseminar/4 — *Baudisch, Patrick*

024 **Large Language Models and Computer Vision Research Seminar**
Projektseminar/4 — *de Melo, Gerard; Zhang, Jingyi*

3 **Network Security in Practice**
Seminar/Praktikum/4 — *Najafi, Peyman; Cheng, Feng*

025 **Computing on Encrypted Data**

Vorlesung/Übung/ 2

This course offers an introduction to cryptographic techniques that enable computation over encrypted data, with a central focus on Homomorphic Encryption. We will follow a practical and engineering-focused approach: while we will touch on essential theoretical concepts, the primary emphasis will be on equipping participants with the skills needed to implement these techniques in real-world applications. The course will comprise a hands-on project where participants will apply what they've learned to develop a functional cryptographic system.

Exam: The grading will be based on a final exam (70%) and a practical project evaluation (30%). The final exam will be oral, unless too many participant register

Content of teaching:

- Definitions and model
- Early constructions
- Current, lattice-based constructions
- Multiparty homomorphic encryption & Secure multiparty computations
- Implementation

Prerequisites:

- Introduction to cryptography: encryption, security property and game-based proofs.
- Basic discrete mathematics: modular algebra, very basic group and ring theory.
- Programming: current HE implementation are in C++ and Go.

Mouchet, Christian
Lehmann, Anja

0 **Mobile Security**

Vorlesung/Übung/ 4

This lecture covers mobile security on an application and system level, with many hands-on exercises. Students will learn state-of-the-art security concepts for both, iOS and Android, and will be able to perform security testing of mobile apps, mobile malware analysis, as well as testing security-critical components within mobile operating systems.

https://moodle.hpi.de/course/view.php?id=798
This lecture covers mobile security on an application and system level, with many hands-on exercises. Students will learn state-of-the-art security concepts for both iOS and Android. They will be able to perform security testing of mobile apps, mobile malware analysis, and testing security-critical components within mobile operating systems.

Course contents include:

- Threat modeling for mobile devices and apps,
- building mobile applications with Xcode and Android Studio,
- application security and testing,
- mobile malware capabilities and detection,
- operating system internals, such as inter-process communication, threads, ...,
- kernel and firmware security,
- mobile forensics, and
- wireless security.

Grading is based on practical exercises and the final exam.

- Assignments (50%)
- Written exam, 90 minutes (50%)

Classen, Jiska

018 **Kryptographie**

Vorlesung/Übung/ 4

Die Vorlesung gibt eine umfassende Einführung in die moderne Kryptographie und die Grundkonzepte der beweisbaren Sicherheit. Dazu werden formale Angreifermodelle definiert und die Sicherheit der vorgestellten Kryptoverfahren unter wohldefinierten Komplexitätsannahmen in diesem Angreifermodell nachgewiesen. Der Vorlesung dient auch als Grundlage für andere Kurse zur Kryptographie, die vom Lehrstuhl angeboten werden.

Content of teaching

- Informationstheoretische vs. Komplexitätstheoretische Sicherheit
- Symmetrische Kryptographie
  - Symmetrische Verschlüsselung
  - Pseudozufallsfunktionen
  - Message Authentication Codes (MAC)
  - Hash-Funktionen
  - Authenticated Encryption
- Asymmetrische Kryptographie
  - Diffie-Hellman Schlüsselaustausch
  - Public-Key Verschlüsselung
  - Digitale Signaturen

Die Vorlesung setzt Grundkenntnisse in Mathematik und theoretischer Informatik voraus, insbesondere müssen die formale mathematische Sprache und elementare Beweistechniken (Widerspruchsbeweis) problemlos angewandt werden können. Wenn diese Kenntnisse nicht vorhanden sind, wird empfohlen dieses Wissen vor der Vorlesung selbstständig zu erwerben, z.B. durch die Teilnahme an den Vorlesungen Mathematik I oder II (ITSE-Bachelor). In den ersten Vorlesungswochen wird es voraussichtlich auch zusätzliche Übungstermine und -materialien geben, in denen elementare Grundlagen aufgefrischt werden können.

Lehmann, Anja
Dayanikli, Dennis Kenan

017 **Digital Entomology: Tracking and Tackling Cyber Bugs**

Seminar/3

Cybersecurity attacks happen frequently and have severe impact. Bugs in digital systems make these attacks possible. In this seminar, we'll take a look into these bugs, why they happen, how they can be exploited, and what could be done to mitigate them. We're collecting and studying cyber bugs – and you'll all be digital entomologists!
https://moodle.hpi.de/course/edit.php?id=799

The seminar follows a weekly schedule. Each week, we'll talk about recent, impactful bugs. The research talks will be split into bugs presented by the lecturer as well as bugs presented by students. We aim at covering highly diverse and recent bugs and bug classes, such as:

- web and browser security,
- internet-facing services including firewalls, mail, ...,
- binary exploitation,
- real-world bugs in cryptographic implementations,
- hardware bugs,
- ...

Students can pick the bugs they present on their own, but there'll be some moderation to ensure no duplicate bugs and a high variety.

Some experience in the area of cyber security is recommended. You should be able to follow technical writeups about bugs and how they were exploited in order to give presentations about these bugs.

Exam

- 70% Presentations (two 30 minute presentations per student – that means two bugs being presented; each presentation is 20 minutes talk + 10 minutes Q&A)
- 30% Creating quizzes (create multiple choice quizzes for two pressentations)
- Passing all multiple choice quizzes during the semester with at least 75% is mandatory, multiple attempts are allowed.

Classen, Jiska

0 **Cyber Security Management**

Vorlesung/Übung/4 — *Doerr, Christian*

6 **Graphenalgorithmen**

Vorlesung/Übung/4 — *Friedrich, Tobias; Skretas, Georgios*

6 **Advanced Competitive Programming 2**

Vorlesung/4 — *Friedrich, Tobias; Simonov, Kirill; Cohen, Sarel*

003 **Understanding Graphs, Algorithms, Randomness**

Seminar/2 — *Friedrich, Tobias; Goebel, Andreas; Verma, Shaily*

005 **Advanced Topics in Algorithms and Complexity**

Vorlesung/4 — *Friedrich, Tobias; Goebel, Andreas; Verma, Shaily*

045 **Algorithms for Collective Decision Making**

Vorlesung/Übung/4 — *Boehmer, Niclas*

This module deals with collective decision making, where a group of agents with preferences over alternatives seeks to select a compromise alternative that fairly reflects everyone's preferences. We focus on three types of collective decision making scenarios:
Voting: Selecting one or more candidates to represent a population of voters based on their preferences over candidates**.**
Resource Allocation: Fairly and efficiently distributing a set of items among agents.
Coalition Formation: Dividing agents into teams based on their preferences for different teams.
The course takes a primarily theoretical approach to these problems, rooted in computational social choice, a field at the intersection of theoretical computer science and economics. We study collective decision making problems from four perspectives, which are all also relevant beyond computational social choice:
Algorithmic: How efficiently can we find a winning alternative?
Axiomatic: Can we design an algorithm that satisfies a set of desirable normative properties?
Game-theoretic: Can agents strategically manipulate the algorithm/outcome?
Experimental: How do different algorithms behave in practice?

The course will consist of three parts: Voting, resource allocation, and coalition formation, where the first part is roughly as long as the other two combined. Covered topics include:
Voting

- Single Winner Voting & Rank Aggregation: voting rules, winner determination problem, axiomatic characterizations and impossibility results, manipulation, robustness, other computational problems around elections
- Multiwinner Voting & Participatory Budgeting: Voting rules, winner determination problem, proportionality axioms, transparency, real-world instances
- Applications: clustering, proof-of-stake blockchain, deliberation, LLMs / reinforcement learning from human feedback

Resource Allocation

- Divisible Goods: fairness axioms, Robertson-Webb model and query complexity, price of proportionality
- Indivisible Goods: fairness axioms, computing fair allocations

Coalition Formation/ Cooperative Game Theory

- Transferable utilities: stability concepts, Shapely value and its applications
- Non-transferable utilities: hedonic games and stable matching, stability concepts, computing stable outcomes

Final Exam: The planned exam mode is a ~30-minute oral exam, which will constitute 100% of the course grade. An average grade of at least 50% in the exercises is required for students to participate in the final exam but does not contribute towards the course grade.
Exercises: Exercises will be assigned on a (bi-)weekly basis and will consist of two types: (1) Traditional problem-solving exercise sheets and (2) Readings of (parts of) research papers, accompanied by comprehension questions.

6 **Trends in Betriebssystemen und Middleware (Forschungsseminar)**

Seminar/2 — *Polze, Andreas*

1 **Mobilkommunikation**

Vorlesung/Übung/4 — For details, please check Moodle. — *Karl, Holger*

8 **Advanced Machine Learning Seminar**

Seminar/4 — *Lippert, Christoph*

019 **Modern and Secure Internet: Design and Operations**

Vorlesung/4 — *Bajpai, Vaibhav*
Ververis, Vasileios

0	**Explaining and Visualizing AI**		
	Seminar/Praktikum /4		*Burmeister, Josafat-Mattias* *Cech, Tim* *Doellner, Juergen*

Spezialisierung (HPI-ISAE-S)

3	**Creating Interactive 3D Web Apps with TypeScript**		
	Projektseminar/4		*Baudisch, Patrick*
024	**Large Language Models and Computer Vision Research Seminar**		
	Projektseminar/4		*de Melo, Gerard* *Zhang, Jingyi*
3	**Network Security in Practice**		
	Seminar/Praktikum /4		*Najafi, Peyman* *Cheng, Feng*
0	**Mobile Security**		
	Vorlesung/Übung/ 4	This lecture covers mobile security on an application and system level, with many hands-on exercises. Students will learn state-of-the-art security concepts for both, iOS and Android, and will be able to perform security testing of mobile apps, mobile malware analysis, as well as testing security-critical components within mobile operating systems. https://moodle.hpi.de/course/view.php?id=798 This lecture covers mobile security on an application and system level, with many hands-on exercises. Students will learn state-of-the-art security concepts for both iOS and Android. They will be able to perform security testing of mobile apps, mobile malware analysis, and testing security-critical components within mobile operating systems. Course contents include: Threat modeling for mobile devices and apps, building mobile applications with Xcode and Android Studio, application security and testing, mobile malware capabilities and detection, operating system internals, such as inter-process communication, threads, ..., kernel and firmware security, mobile forensics, and wireless security. Grading is based on practical exercises and the final exam. Assignments (50%) Written exam, 90 minutes (50%)	*Classen, Jiska*

017 **Digital Entomology: Tracking and Tackling Cyber Bugs**

Seminar/3

Cybersecurity attacks happen frequently and have severe impact. Bugs in digital systems make these attacks possible. In this seminar, we'll take a look into these bugs, why they happen, how they can be exploited, and what could be done to mitigate them. We're collecting and studying cyber bugs – and you'll all be digital entomologists!
https://moodle.hpi.de/course/edit.php?id=799

The seminar follows a weekly schedule. Each week, we'll talk about recent, impactful bugs. The research talks will be split into bugs presented by the lecturer as well as bugs presented by students. We aim at covering highly diverse and recent bugs and bug classes, such as:

- web and browser security,
- internet-facing services including firewalls, mail, ...,
- binary exploitation,
- real-world bugs in cryptographic implementations,
- hardware bugs,
- ...

Students can pick the bugs they present on their own, but there'll be some moderation to ensure no duplicate bugs and a high variety.

Some experience in the area of cyber security is recommended. You should be able to follow technical writeups about bugs and how they were exploited in order to give presentations about these bugs.

Exam

- 70% Presentations (two 30 minute presentations per student – that means two bugs being presented; each presentation is 20 minutes talk + 10 minutes Q&A)
- 30% Creating quizzes (create multiple choice quizzes for two pressentations)
- Passing all multiple choice quizzes during the semester with at least 75% is mandatory, multiple attempts are allowed.

Classen, Jiska

003 **Understanding Graphs, Algorithms, Randomness**

Seminar/2

Friedrich, Tobias
Goebel, Andreas
Verma, Shaily

005 **Advanced Topics in Algorithms and Complexity**

Vorlesung/4

Friedrich, Tobias
Goebel, Andreas
Verma, Shaily

045 **Algorithms for Collective Decision Making**

Vorlesung/Übung/4 — *Boehmer, Niclas*

This module deals with collective decision making, where a group of agents with preferences over alternatives seeks to select a compromise alternative that fairly reflects everyone's preferences. We focus on three types of collective decision making scenarios:
Voting: Selecting one or more candidates to represent a population of voters based on their preferences over candidates**.**
Resource Allocation: Fairly and efficiently distributing a set of items among agents.
Coalition Formation: Dividing agents into teams based on their preferences for different teams.
The course takes a primarily theoretical approach to these problems, rooted in computational social choice, a field at the intersection of theoretical computer science and economics. We study collective decision making problems from four perspectives, which are all also relevant beyond computational social choice:
Algorithmic: How efficiently can we find a winning alternative?
Axiomatic: Can we design an algorithm that satisfies a set of desirable normative properties?
Game-theoretic: Can agents strategically manipulate the algorithm/outcome?
Experimental: How do different algorithms behave in practice?

The course will consist of three parts: Voting, resource allocation, and coalition formation, where the first part is roughly as long as the other two combined. Covered topics include:
Voting

- Single Winner Voting & Rank Aggregation: voting rules, winner determination problem, axiomatic characterizations and impossibility results, manipulation, robustness, other computational problems around elections
- Multiwinner Voting & Participatory Budgeting: Voting rules, winner determination problem, proportionality axioms, transparency, real-world instances
- Applications: clustering, proof-of-stake blockchain, deliberation, LLMs / reinforcement learning from human feedback

Resource Allocation

- Divisible Goods: fairness axioms, Robertson-Webb model and query complexity, price of proportionality
- Indivisible Goods: fairness axioms, computing fair allocations

Coalition Formation/ Cooperative Game Theory

- Transferable utilities: stability concepts, Shapely value and its applications
- Non-transferable utilities: hedonic games and stable matching, stability concepts, computing stable outcomes

Final Exam: The planned exam mode is a ~30-minute oral exam, which will constitute 100% of the course grade. An average grade of at least 50% in the exercises is required for students to participate in the final exam but does not contribute towards the course grade.
Exercises: Exercises will be assigned on a (bi-)weekly basis and will consist of two types: (1) Traditional problem-solving exercise sheets and (2) Readings of (parts of) research papers, accompanied by comprehension questions.

6 **Trends in Betriebssystemen und Middleware (Forschungsseminar)**

Seminar/2 — *Polze, Andreas*

1 **Mobilkommunikation**

Vorlesung/Übung/4 — For details, please check Moodle. — *Karl, Holger*

8 **Advanced Machine Learning Seminar**

Seminar/4 — *Lippert, Christoph*

019 **Modern and Secure Internet: Design and Operations**

Vorlesung/4 — *Bajpai, Vaibhav*
Ververis, Vasileios

0	**Explaining and Visualizing AI**	
	Seminar/Praktikum /4	*Burmeister, Josafat-Mattias* *Cech, Tim* *Doellner, Juergen*

Operating Systems and Information Systems Technology (OSIS)

Konzepte und Methoden (HPI-OSIS-K)

024	**Large Language Models and Computer Vision Research Seminar**	
	Projektseminar/4	*de Melo, Gerard* *Zhang, Jingyi*
3	**Network Security in Practice**	
	Seminar/Praktikum /4	*Najafi, Peyman* *Cheng, Feng*
027	**Process Mining**	
	Vorlesung/Übung/ 2	*Leopold, Henrik* *Weske, Mathias*
0	**Cyber Security Management**	
	Vorlesung/Übung/ 4	*Doerr, Christian*

035 **Advanced Topics in Software Engineering: Automation and AI**

Vorlesung/4

Giese, Holger
Barkowsky, Matthias
Adriano, Christian
Ghahremani, Sona

In software engineering, like many other engineering disciplines, we on the one hand want to build solutions with the best possible quality while we on the other hand must adhere to predetermined budget and time constraints. Furthermore, often not enough qualified software engineers are available. Therefore, improving the productivity during software development and operation as well as the quality of the outcomes by automating activities partially or completely using software itself has been a major area for innovations since the early days of programming and software engineering.
Nowadays, many software engineering activities benefit from a high degree of automation and very often we take that automation for granted and are hardly aware of it anymore. Also, often the considered software systems have become so complex that they can only be developed, operated, and evolved by using largely automated approaches for various software engineering activities.
Automation in software engineering has the goal to partially or fully execute software engineering activities with minimal human intervention, thereby significantly increasing both quality and productivity. Automation successfully encompasses a broad range of activities, for instance, requirements definition, specification, software architecture, software design and synthesis, implementation, modeling, testing, quality assurance, verification, validation, maintenance, evolution, configuration management, deployment, reengineering, reuse and visualization. (…)
Also artificial intelligence is nowadays used to enhance existing software systems or make new beforehand not feasible software systems possible. Therefore, software engineering activities and outcomes have to be adjusted so that software solutions can benefit from integrated features realized with artificial intelligence. This requires that a clear strategy on how to use artificial intelligence in a software is established and that all aspects of software development and operation are appropriately adjusted to ensure that the employed combination of traditional software and artificial intelligence results in the required quality.
Therefore, we will look in this course at first into the advanced development of systems using automation for software engineering including artificial Intelligence as well as secondly into software engineering for the development of advanced systems that employ artificial intelligence. Furthermore, we will also investigate the operation of systems and how automation and in particular artificial intelligence can help there. Finally, we will discuss the case where automation and in particular artificial intelligence is used for development and operation and employed for the system itself at the same time.
We will in addition to the discussions in the lecture explore the key challenges also with small projects in the exercises and will collect at the beginning of the course suggestions for artificial intelligence tools to consider for the small projects or student presentations.
1. https://www.infoworld.com/article/3489925/github-survey-finds-nearly-all-developers-using-ai-coding-tools.html
2. https://research.google/blog/ai-in-software-engineering-at-google-progress-and-the-path-ahead/

Exam:
The grading process takes into account two components:
The results of the hands-on projects accompanying the lecture, with each project graded individually.
A final exam at the end of the semester. Depending on the number of course participants, the exam will either be oral or written.
Students will be required to pass both graded components. In particular, completing all hands-on projects to an adequate level is required for admission to the exam.
The final grade will either be composed of the average project grade (50%) and the exam grade (50%) OR the exam grade (100%) only, depending on which grading scheme yields a better result for each student individually.

036 **Software Engineering with Machine Learning: Tools and Methods**

Projektseminar/4

We will grade the group's paper report (80%) and presentations (20%). Note that the report includes documenting the experiments and the obtained results. Therefore, the grading of the report includes the experiments. During the project phase, we will require participation in meetings and other groups' presentations in the form of questions and feedback to their peers.

In the field of software engineering, the need to balance quality, budget constraints, and time limitations are constant drivers for innovation in tools and methods. Because software engineering tasks are extremely labor intensive, automation has become a critical area of focus, aiming to improve productivity during software development and operation while maintaining high-quality code and specificaitons. As a result, many software engineering tasks currently benefit from automation. Meanwhile, artificial intelligence (AI) in general and various specific Machine Learning methods have been bringing new opportunities for automation.. Even before the term “software engineering” was coined, AI was considered a candidate technology. Currently, AI is poised to revolutionize software development. Surveys show that over 97% of developers have used AI coding tools, and companies like Google already produce 50% of their code using AI. AI enhances existing software systems and enables previously unfeasible solutions. However, a clear strategy is essential to integrate AI effectively, adjusting all aspects of software development and operation to ensure the desired quality.
Finally, in this project seminar, we will develop projects that explore how to advance software engineering tasks using automation and specific machine learning methods, from Large Language Models to Reinforcement Learning and Graph Neural Networks. We will also discuss in the context of the projects the particularities of software engineering for AI-driven systems and how automation and AI impact system operation.
This project seminar is a companion of the course “Advanced Topics in Software Engineering: Automation and AI (ASE)”, in a sense that the conceptual and theoretical topics will be covered in the lecture, while the project seminar will focus on more in-depth designs and prototypes. For this reason the participants in the project seminar are invited to attend the ASE lectures.

Barkowsky, Matthias
Giese, Holger
Adriano, Christian

029 **Modeling of Embedded Systems using Graphtransformation**

Projektseminar/4

Embedded systems consist of software components that observe and control a physical environment. The discrete parts of the states of such embedded systems can be represented by graphs. The behavior of such embedded systems can then be described by various kinds of graph transformation systems capturing aspects such as time and probabilism at varying levels of detail. In this course we use tools to model and analyze embedded systems using graph transformation systems.

The course begins with an introduction to graphs, graph transformation steps, and graph transformation systems. Students will then work in groups to understand the concepts presented though manual calculation, implementation, and use of tools. Students will submit and present the results of each phase.
Phase 1: Graph transformation fundamentals.
Phase 2: Graph transformation modeling using the Groove tool, followed by analysis.
Phase 3: Graph transformation modeling using probabilistic timed graph transformation systems using the Henshin tools, followed by analysis.
Moodle Course

Exam
Modulprüfungen: Mündliche Prüfung, 30-45 Minuten
Prüfungsnebenleistungen: Für die Zulassung zur Modulprüfung: Übungsaufgaben (50%)

Giese, Holger
Maximova, Maria
Schneider, Sven

5 **Global Team-Based Innovation I**

Projektseminar/4

Uebernickel, Falk
Beermann, Vincent
Enkmann, Jan
Rolfes, Theresa Maria
Cauderay, Virginie
Wuttke, Tobias

Global Team-based Innovation (GTI) is a course designated for master students of the Hasso Plattner Institute (HPI) and the University of Potsdam (UP).
In our course, students apply IT knowledge to engineer digital solutions for real business challenges provided by prominent global companies. We follow the Design Thinking methodology to innovate on wicked problems given by our project partners. Within GTI, HPI students collaborate with students from other leading global universities: HPI is a partner in ME310 (for projects with the Stanford University) as well as part of the SUGAR Network for Design Innovation (for projects with other global universities).
https://hpi.de/uebernickel/teaching/global-team-based-innovation-gti-design-thinking.html
This class is exclusively available to students who have been accepted through our application process.

Exam
- Project work (20%)
  - Individual participation during lectures, group meetings and in project work
  - Stakeholder management
  - Project management (sticking to deadlines, etc.)
- Milestone presentations (20%)
  - GTI 1: Fall & winter presentation
  - GTI 2: Final presentation
- Tangible outcomes (20%)
  - One-Pagers for corporate partners
  - Intermediate prototypes
- Milestone documentations (40%)
  - GTI 1: Fall & winter documentation
  - GTI 2: Final documentation & videos

The estimated workload is 2-3 days per week.

Goals:
Students from Potsdam and leading global partner universities tackle design innovation challenges posed by global corporations. The 9 months (2 semesters) course focuses on the application of IT knowledge for engineering solutions to real business challenges. Further, we put emphasis on teaching students human-centered innovation methods and processes required for designers, engineers, and project managers of the future.
Within the projects, students go through an intense and iterative process of need finding, ideation, and rapid prototyping to create and evaluate new concepts. Company involvement provides the reality check necessary for teams to improve their innovation abilities. The team is supported by a professional coach, corporate liaisons, and faculty advisors.
Projects typically involve systems integration and include a mix of mechanical, electronic and software design. The results of all projects are real prototypes that have a user-centric design, are economically viable and technically feasible.

9 **Applied Probabilistic Machine Learning**

Seminar/4

Richard, Hugues
Renard, Bernhard Yves

028 **Deep Learning for Molecular Biology**

Seminar/2

Renard, Bernhard Yves
Rissom, Francesca
Heyne, Henrike
Nowicka, Melania Maria
Bartoszewicz, Jakub
Maciej

Rapid advances in both biology—through increased data availability and the insights derived from it—and in methods for handling high-dimensional data, such as deep learning architectures and computational resources, have created exciting opportunities for integrating these fields.

This seminar will examine how state-of-the-art deep learning models, including **CNNs, GNNs, Transformers,** and **Diffusion models**, are applied to **genome, RNA,** and **protein sequence** analysis.
We will explore how these advances are used to address key questions such as the effects of genetic mutations, protein structure and function prediction, and the design of new molecules for therapeutic purposes.
The course will primarily consist of **student presentations** on recent, preselected publications in these areas, followed by in-depth **discussions**.

Biological background is **not** necessary to participate in the seminar, but you will need a basic understanding of deep learning.
Good **English** skills are required to understand and discuss current literature.

In the seminar, each participant will give a presentation about a predefined topic within the research area and a short report. The final grade consists of the following parts:

- Oral presentation (60%)
- Written report (30%)
- Participation (10%)

Goals:

- Identify **current topics** and **open challenges** in the field of artificial intelligence for molecular biology
- Improve your understanding of **best practices in scientific research**
- **Effectively communicate** complex scientific topics in this field and lead a discussion
- Improving **presentation** and **writing skills**

The first three sessions will be in lecture format, providing an introduction to key biological concepts and a refresher on deep learning architectures. Following these sessions, students will give oral presentations on select scientific articles including a brief introduction to specific topics. These articles can be chosen from a list that will be presented during the initial meetings.
The seminar will be conducted on-site (with a hybrid option if needed).
Please register on the course's Moodle page for further information.

Max. number of participants: 10

7 **Advanced Data Profiling**

Projektseminar/4 **Data Profiling for Dynamic Data** *Naumann, Felix*
Kaminsky, Youri
Lindner, Daniel
Schmidl, Sebastian

https://hpi.de/naumann/teaching/current-courses/ws-24-25/advanced-data-profiling.html

Data profiling is the process of extracting metadata from datasets [1]. Researchers have proposed plenty of profiling algorithms for all different kinds of data dependencies, such as Unique Column Combinations (UCCs), Functional Dependencies (FDs), Inclusion Dependencies (INDs), or Order Dependencies (ODs), on static data in a batch process. However, many real-world datasets are constantly changing. These changes, which are inserts, updates, and deletes, also change the datasets' metadata, making it necessary to frequently re-profile the data. Unfortunately, executing the static profiling algorithms on every dataset change is excessively expense — even infeasible — because the static approaches do not leverage the knowledge about an earlier state of the dataset and its dependencies. This calls for novel incremental discovery algorithms that re-use existing profiling results to efficiently maintain data dependencies for dynamic datasets.
We will start with existing solutions to this problem for the following dependency types (depending on the number of students) and then improve upon them:

- **UCCs**: SWAN [2]
- **FDs**: DynFD [3], DHSFD [4]
- **INDs**: Shaabani's algorithm [5]
- **ODs**: list-based: IncOD [6], pointwise: IncPOD [7]

Seminar Organization
We will form teams of two students each. Every team works on one kind of data dependency. First, the teams become familiar with related work as an inspiration. Afterward, each student team develops their own ideas to profile their dependency type.
The students turn their ideas into working algorithms. There are two main goals for each algorithm:
1) The complete set of minimal or maximal dependencies must be maintained.
2) The runtime of the algorithm is to be optimized.
Datasets for benchmarking are provided to the students. Finally, the students present their approaches and write a short report.

Prior knowledge in data profiling (preferably completed Data Profiling lecture)
Good programming skills in a major programming language

013 **DQ4AI: Data Quality Assessment**

Projektseminar/4 *Naumann, Felix*
Ehrlinger, Lisa
Mohammed, Sedir

015 **Table Representation Learning**

Projektseminar/4

Naumann, Felix
Laskowski, Lukas
Pugnaloni, Francesco
Hoenes, Christoph

Representation learning (RL) aims to find meaningful representations of given objects to make them easier to process or understand. It finds application in various areas, e.g., cybersecurity, healthcare, time-series analysis, natural language processing, audio processing, and table understanding, and can be used to process data in different modalities, e.g., images, text, audio, or tabular data.
After the rise of foundation models, finding compact and uniform representations of different modalities of data became more important than ever, but while text and images have strong and consolidated representation methods, tabular data have been overlooked until recently. The research area that is trying to fill this gap is called table representation learning (TRL) and aims to extract meaningful information from tabular data to create expressive vectorial representations.

In this seminar, we will introduce you to the field of table representation learning, and explore together how different approaches perform in classic table-related tasks. To achieve that, we have the following plan:

Team activities: each team ideally consists of 2 students and will be assigned a specific TRL archetype, e.g., graph-based, LLM-based, word-embedding-based, etc. Your part is to choose one or more representative models from the ones proposed, implement them, and use them to solve classic table-related tasks, e.g., entity resolution, schema matching, etc.
Deliverable: The outcome of the seminar is a paper-style technical report that the teams will write collaboratively to present the results of the conducted analysis. In addition to the code, models, and datasets that have been produced.
Bonus: You will learn how to read/write a research paper and how to conduct scientific experiments and present the results in a paper.

Prerequisites:

- Python
- Basic knowledge of machine learning and deep learning

Organization
The organizational details for this seminar are as follows:

- Project seminar for master students
- Language of instruction: English
- 6 credit points, 4 SWS
- At most 6 participants (ideally, 3 teams of 2 students each)

Grading
In the seminar, each team will develop an approach and write a short report. The final grade consists of the following three parts:

- Approach (35%)
- Written report (35%)
- Midterm and final presentations (30%)

6 **Trends in Betriebssystemen und Middleware (Forschungsseminar)**

Seminar/2 *Polze, Andreas*

1 **Mobilkommunikation**

Vorlesung/Übung/4 For details, please check Moodle. *Karl, Holger*

002 **Digital Health and Research Systems, Data Interoperability**

Vorlesung/Seminar/4 *Heitmann, Kai U.*
Thun, Sylvia
Prasser, Fabian
Arnrich, Bert

5 **Biostatistics & Epidemiological data analysis using R**

Vorlesung/4 *Konigorski, Stefan*

4 **Big Data Systeme**

Vorlesung/4 *Rabl, Tilmann*
Boissier, Martin
Salazar Diaz, Ricardo
Strassenburg, Nils

021 **Machine Learning Systems**
Projektseminar/4 — *Rabl, Tilmann; Salazar Diaz, Ricardo; Strassenburg, Nils; Tolovski, Ilin*

020 **Data Processing on Modern Hardware**
Projektseminar/4 — *Rabl, Tilmann; Weisgut, Marcel*

019 **Modern and Secure Internet: Design and Operations**
Vorlesung/4 — *Bajpai, Vaibhav; Ververis, Vasileios*

6 **Build Your Own Programming Language**
Vorlesung/Seminar/4 — *Hirschfeld, Robert; Lincke, Jens; Felgentreff, Tim; Niephaus, Fabio*

Programming languages and how they work sometimes feel like magic, and the people who create those arcane technologies are often treated like wizards. In this course, students will dispel this magic and learn how to build a programming language themselves.

There will be a combined seminar/lecture every week. Every student has to continously work on the implemententation of their language and show progress every week.

- In-depth knowledge in at least one dynamic programming language
- Knowledge of Java and associated technologies helpful, but not required

Grading will take place based on the continuous work on the projects and the final oral examination. To complete the course, the following requirements are to be fulfilled, and the grade will be composed of:

- Regular submission of implementation progress (weekly) (20%)
- Functional implementation of the language at the end of the semester (30%)
- Oral exam at end of semester (50%)
- Bonus Points from weekly challenges

All source code created during this seminar will be licenced under the MIT license

Oral exam at end of semester

2 **Spatial Data: Processing and Visualization Techniques**
Seminar/Praktikum/4 — *Richter, Rico; Wegen, Ole; Hildebrand, Justus; Schulz, Sebastian; Burmeister, Josafat-Mattias*

Technolgoien und Werkzeuge (HPI-OSIS-T)

024 **Large Language Models and Computer Vision Research Seminar**
Projektseminar/4 — *de Melo, Gerard; Zhang, Jingyi*

3 **Network Security in Practice**
Seminar/Praktikum/4 — *Najafi, Peyman; Cheng, Feng*

027 **Process Mining**
Vorlesung/Übung/2 — *Leopold, Henrik; Weske, Mathias*

0 **Cyber Security Management**
Vorlesung/Übung/4 — *Doerr, Christian*

035 **Advanced Topics in Software Engineering: Automation and AI**

Vorlesung/4

Giese, Holger
Barkowsky, Matthias
Adriano, Christian
Ghahremani, Sona

In software engineering, like many other engineering disciplines, we on the one hand want to build solutions with the best possible quality while we on the other hand must adhere to predetermined budget and time constraints. Furthermore, often not enough qualified software engineers are available. Therefore, improving the productivity during software development and operation as well as the quality of the outcomes by automating activities partially or completely using software itself has been a major area for innovations since the early days of programming and software engineering.
Nowadays, many software engineering activities benefit from a high degree of automation and very often we take that automation for granted and are hardly aware of it anymore. Also, often the considered software systems have become so complex that they can only be developed, operated, and evolved by using largely automated approaches for various software engineering activities.
Automation in software engineering has the goal to partially or fully execute software engineering activities with minimal human intervention, thereby significantly increasing both quality and productivity. Automation successfully encompasses a broad range of activities, for instance, requirements definition, specification, software architecture, software design and synthesis, implementation, modeling, testing, quality assurance, verification, validation, maintenance, evolution, configuration management, deployment, reengineering, reuse and visualization. (…)
Also artificial intelligence is nowadays used to enhance existing software systems or make new beforehand not feasible software systems possible. Therefore, software engineering activities and outcomes have to be adjusted so that software solutions can benefit from integrated features realized with artificial intelligence. This requires that a clear strategy on how to use artificial intelligence in a software is established and that all aspects of software development and operation are appropriately adjusted to ensure that the employed combination of traditional software and artificial intelligence results in the required quality.
Therefore, we will look in this course at first into the advanced development of systems using automation for software engineering including artificial Intelligence as well as secondly into software engineering for the development of advanced systems that employ artificial intelligence. Furthermore, we will also investigate the operation of systems and how automation and in particular artificial intelligence can help there. Finally, we will discuss the case where automation and in particular artificial intelligence is used for development and operation and employed for the system itself at the same time.
We will in addition to the discussions in the lecture explore the key challenges also with small projects in the exercises and will collect at the beginning of the course suggestions for artificial intelligence tools to consider for the small projects or student presentations.
1. https://www.infoworld.com/article/3489925/github-survey-finds-nearly-all-developers-using-ai-coding-tools.html
2. https://research.google/blog/ai-in-software-engineering-at-google-progress-and-the-path-ahead/

Exam:
The grading process takes into account two components:
The results of the hands-on projects accompanying the lecture, with each project graded individually.
A final exam at the end of the semester. Depending on the number of course participants, the exam will either be oral or written.
Students will be required to pass both graded components. In particular, completing all hands-on projects to an adequate level is required for admission to the exam.
The final grade will either be composed of the average project grade (50%) and the exam grade (50%) OR the exam grade (100%) only, depending on which grading scheme yields a better result for each student individually.

036 **Software Engineering with Machine Learning: Tools and Methods**

Projektseminar/4

We will grade the group's paper report (80%) and presentations (20%). Note that the report includes documenting the experiments and the obtained results. Therefore, the grading of the report includes the experiments. During the project phase, we will require participation in meetings and other groups' presentations in the form of questions and feedback to their peers.

In the field of software engineering, the need to balance quality, budget constraints, and time limitations are constant drivers for innovation in tools and methods. Because software engineering tasks are extremely labor intensive, automation has become a critical area of focus, aiming to improve productivity during software development and operation while maintaining high-quality code and specificaitons. As a result, many software engineering tasks currently benefit from automation. Meanwhile, artificial intelligence (AI) in general and various specific Machine Learning methods have been bringing new opportunities for automation.. Even before the term “software engineering” was coined, AI was considered a candidate technology. Currently, AI is poised to revolutionize software development. Surveys show that over 97% of developers have used AI coding tools, and companies like Google already produce 50% of their code using AI. AI enhances existing software systems and enables previously unfeasible solutions. However, a clear strategy is essential to integrate AI effectively, adjusting all aspects of software development and operation to ensure the desired quality.
Finally, in this project seminar, we will develop projects that explore how to advance software engineering tasks using automation and specific machine learning methods, from Large Language Models to Reinforcement Learning and Graph Neural Networks. We will also discuss in the context of the projects the particularities of software engineering for AI-driven systems and how automation and AI impact system operation.
This project seminar is a companion of the course “Advanced Topics in Software Engineering: Automation and AI (ASE)”, in a sense that the conceptual and theoretical topics will be covered in the lecture, while the project seminar will focus on more in-depth designs and prototypes. For this reason the participants in the project seminar are invited to attend the ASE lectures.

Barkowsky, Matthias
Giese, Holger
Adriano, Christian

029 **Modeling of Embedded Systems using Graphtransformation**

Projektseminar/4

Embedded systems consist of software components that observe and control a physical environment. The discrete parts of the states of such embedded systems can be represented by graphs. The behavior of such embedded systems can then be described by various kinds of graph transformation systems capturing aspects such as time and probabilism at varying levels of detail. In this course we use tools to model and analyze embedded systems using graph transformation systems.

The course begins with an introduction to graphs, graph transformation steps, and graph transformation systems. Students will then work in groups to understand the concepts presented though manual calculation, implementation, and use of tools. Students will submit and present the results of each phase.
Phase 1: Graph transformation fundamentals.
Phase 2: Graph transformation modeling using the Groove tool, followed by analysis.
Phase 3: Graph transformation modeling using probabilistic timed graph transformation systems using the Henshin tools, followed by analysis.
Moodle Course

Exam
Modulprüfungen: Mündliche Prüfung, 30-45 Minuten
Prüfungsnebenleistungen: Für die Zulassung zur Modulprüfung: Übungsaufgaben (50%)

Giese, Holger
Maximova, Maria
Schneider, Sven

5 **Global Team-Based Innovation I**

Projektseminar/4

Uebernickel, Falk
Beermann, Vincent
Enkmann, Jan
Rolfes, Theresa Maria
Cauderay, Virginie
Wuttke, Tobias

Global Team-based Innovation (GTI) is a course designated for master students of the Hasso Plattner Institute (HPI) and the University of Potsdam (UP).
In our course, students apply IT knowledge to engineer digital solutions for real business challenges provided by prominent global companies. We follow the Design Thinking methodology to innovate on wicked problems given by our project partners. Within GTI, HPI students collaborate with students from other leading global universities: HPI is a partner in ME310 (for projects with the Stanford University) as well as part of the SUGAR Network for Design Innovation (for projects with other global universities).
https://hpi.de/uebernickel/teaching/global-team-based-innovation-gti-design-thinking.html
This class is exclusively available to students who have been accepted through our application process.

Exam
- Project work (20%)
  - Individual participation during lectures, group meetings and in project work
  - Stakeholder management
  - Project management (sticking to deadlines, etc.)
- Milestone presentations (20%)
  - GTI 1: Fall & winter presentation
  - GTI 2: Final presentation
- Tangible outcomes (20%)
  - One-Pagers for corporate partners
  - Intermediate prototypes
- Milestone documentations (40%)
  - GTI 1: Fall & winter documentation
  - GTI 2: Final documentation & videos

The estimated workload is 2-3 days per week.

Goals:
Students from Potsdam and leading global partner universities tackle design innovation challenges posed by global corporations. The 9 months (2 semesters) course focuses on the application of IT knowledge for engineering solutions to real business challenges. Further, we put emphasis on teaching students human-centered innovation methods and processes required for designers, engineers, and project managers of the future.
Within the projects, students go through an intense and iterative process of need finding, ideation, and rapid prototyping to create and evaluate new concepts. Company involvement provides the reality check necessary for teams to improve their innovation abilities. The team is supported by a professional coach, corporate liaisons, and faculty advisors.
Projects typically involve systems integration and include a mix of mechanical, electronic and software design. The results of all projects are real prototypes that have a user-centric design, are economically viable and technically feasible.

9 **Applied Probabilistic Machine Learning**

Seminar/4

Richard, Hugues
Renard, Bernhard Yves

028 **Deep Learning for Molecular Biology**

Seminar/2

Renard, Bernhard Yves
Rissom, Francesca
Heyne, Henrike
Nowicka, Melania Maria
Bartoszewicz, Jakub
Maciej

Rapid advances in both biology—through increased data availability and the insights derived from it—and in methods for handling high-dimensional data, such as deep learning architectures and computational resources, have created exciting opportunities for integrating these fields.

This seminar will examine how state-of-the-art deep learning models, including **CNNs, GNNs, Transformers,** and **Diffusion models**, are applied to **genome, RNA,** and **protein sequence** analysis.
We will explore how these advances are used to address key questions such as the effects of genetic mutations, protein structure and function prediction, and the design of new molecules for therapeutic purposes.
The course will primarily consist of **student presentations** on recent, preselected publications in these areas, followed by in-depth **discussions**.

Biological background is **not** necessary to participate in the seminar, but you will need a basic understanding of deep learning.
Good **English** skills are required to understand and discuss current literature.

In the seminar, each participant will give a presentation about a predefined topic within the research area and a short report. The final grade consists of the following parts:

- Oral presentation (60%)
- Written report (30%)
- Participation (10%)

Goals:

- Identify **current topics** and **open challenges** in the field of artificial intelligence for molecular biology
- Improve your understanding of **best practices in scientific research**
- **Effectively communicate** complex scientific topics in this field and lead a discussion
- Improving **presentation** and **writing skills**

The first three sessions will be in lecture format, providing an introduction to key biological concepts and a refresher on deep learning architectures. Following these sessions, students will give oral presentations on select scientific articles including a brief introduction to specific topics. These articles can be chosen from a list that will be presented during the initial meetings.
The seminar will be conducted on-site (with a hybrid option if needed).
Please register on the course's Moodle page for further information.

Max. number of participants: 10

7 **Advanced Data Profiling**

Projektseminar/4

Data Profiling for Dynamic Data
https://hpi.de/naumann/teaching/current-courses/ws-24-25/advanced-data-profiling.html

Naumann, Felix
Kaminsky, Youri
Lindner, Daniel
Schmidl, Sebastian

Data profiling is the process of extracting metadata from datasets [1]. Researchers have proposed plenty of profiling algorithms for all different kinds of data dependencies, such as Unique Column Combinations (UCCs), Functional Dependencies (FDs), Inclusion Dependencies (INDs), or Order Dependencies (ODs), on static data in a batch process. However, many real-world datasets are constantly changing. These changes, which are inserts, updates, and deletes, also change the datasets' metadata, making it necessary to frequently re-profile the data. Unfortunately, executing the static profiling algorithms on every dataset change is excessively expense — even infeasible — because the static approaches do not leverage the knowledge about an earlier state of the dataset and its dependencies. This calls for novel incremental discovery algorithms that re-use existing profiling results to efficiently maintain data dependencies for dynamic datasets.
We will start with existing solutions to this problem for the following dependency types (depending on the number of students) and then improve upon them:

- **UCCs**: SWAN [2]
- **FDs**: DynFD [3], DHSFD [4]
- **INDs**: Shaabani's algorithm [5]
- **ODs**: list-based: IncOD [6], pointwise: IncPOD [7]

Seminar Organization
We will form teams of two students each. Every team works on one kind of data dependency. First, the teams become familiar with related work as an inspiration. Afterward, each student team develops their own ideas to profile their dependency type.
The students turn their ideas into working algorithms. There are two main goals for each algorithm:
1) The complete set of minimal or maximal dependencies must be maintained.
2) The runtime of the algorithm is to be optimized.
Datasets for benchmarking are provided to the students. Finally, the students present their approaches and write a short report.

Prior knowledge in data profiling (preferably completed Data Profiling lecture)
Good programming skills in a major programming language

013 **DQ4AI: Data Quality Assessment**

Projektseminar/4

Naumann, Felix
Ehrlinger, Lisa
Mohammed, Sedir

015 **Table Representation Learning**

Projektseminar/4

Representation learning (RL) aims to find meaningful representations of given objects to make them easier to process or understand. It finds application in various areas, e.g., cybersecurity, healthcare, time-series analysis, natural language processing, audio processing, and table understanding, and can be used to process data in different modalities, e.g., images, text, audio, or tabular data.
After the rise of foundation models, finding compact and uniform representations of different modalities of data became more important than ever, but while text and images have strong and consolidated representation methods, tabular data have been overlooked until recently. The research area that is trying to fill this gap is called table representation learning (TRL) and aims to extract meaningful information from tabular data to create expressive vectorial representations.

In this seminar, we will introduce you to the field of table representation learning, and explore together how different approaches perform in classic table-related tasks. To achieve that, we have the following plan:

Team activities: each team ideally consists of 2 students and will be assigned a specific TRL archetype, e.g., graph-based, LLM-based, word-embedding-based, etc. Your part is to choose one or more representative models from the ones proposed, implement them, and use them to solve classic table-related tasks, e.g., entity resolution, schema matching, etc.
Deliverable: The outcome of the seminar is a paper-style technical report that the teams will write collaboratively to present the results of the conducted analysis. In addition to the code, models, and datasets that have been produced.
Bonus: You will learn how to read/write a research paper and how to conduct scientific experiments and present the results in a paper.

Prerequisites:

- Python
- Basic knowledge of machine learning and deep learning

Organization
The organizational details for this seminar are as follows:

- Project seminar for master students
- Language of instruction: English
- 6 credit points, 4 SWS
- At most 6 participants (ideally, 3 teams of 2 students each)

Grading
In the seminar, each team will develop an approach and write a short report. The final grade consists of the following three parts:

- Approach (35%)
- Written report (35%)
- Midterm and final presentations (30%)

Naumann, Felix
Laskowski, Lukas
Pugnaloni, Francesco
Hoenes, Christoph

6 **Trends in Betriebssystemen und Middleware (Forschungsseminar)**

Seminar/2 *Polze, Andreas*

1 **Mobilkommunikation**

Vorlesung/Übung/4 For details, please check Moodle. *Karl, Holger*

002 **Digital Health and Research Systems, Data Interoperability**

Vorlesung/Seminar/4 *Heitmann, Kai U.*
Thun, Sylvia
Prasser, Fabian
Arnrich, Bert

5 **Biostatistics & Epidemiological data analysis using R**

Vorlesung/4 *Konigorski, Stefan*

4 **Big Data Systeme**

Vorlesung/4 *Rabl, Tilmann*
Boissier, Martin
Salazar Diaz, Ricardo
Strassenburg, Nils

021	**Machine Learning Systems**		
	Projektseminar/4		*Rabl, Tilmann* *Salazar Diaz, Ricardo* *Strassenburg, Nils* *Tolovski, Ilin*
020	**Data Processing on Modern Hardware**		
	Projektseminar/4		*Rabl, Tilmann* *Weisgut, Marcel*
019	**Modern and Secure Internet: Design and Operations**		
	Vorlesung/4		*Bajpai, Vaibhav* *Ververis, Vasileios*
6	**Build Your Own Programming Language**		
	Vorlesung/Seminar/4	Programming languages and how they work sometimes feel like magic, and the people who create those arcane technologies are often treated like wizards. In this course, students will dispel this magic and learn how to build a programming language themselves. There will be a combined seminar/lecture every week. Every student has to continously work on the implemententation of their language and show progress every week. • In-depth knowledge in at least one dynamic programming language • Knowledge of Java and associated technologies helpful, but not required Grading will take place based on the continuous work on the projects and the final oral examination. To complete the course, the following requirements are to be fulfilled, and the grade will be composed of: • Regular submission of implementation progress (weekly) (20%) • Functional implementation of the language at the end of the semester (30%) • Oral exam at end of semester (50%) • Bonus Points from weekly challenges All source code created during this seminar will be licenced under the MIT license Oral exam at end of semester	*Hirschfeld, Robert* *Lincke, Jens* *Felgentreff, Tim* *Niephaus, Fabio*
2	**Spatial Data: Processing and Visualization Techniques**		
	Seminar/Praktikum/4		*Richter, Rico* *Wegen, Ole* *Hildebrand, Justus* *Schulz, Sebastian* *Burmeister, Josafat-Mattias*

Spezialisierung (HPI-OSIS-S)

024	**Large Language Models and Computer Vision Research Seminar**		
	Projektseminar/4		*de Melo, Gerard* *Zhang, Jingyi*
3	**Network Security in Practice**		
	Seminar/Praktikum/4		*Najafi, Peyman* *Cheng, Feng*
027	**Process Mining**		
	Vorlesung/Übung/2		*Leopold, Henrik* *Weske, Mathias*

035 **Advanced Topics in Software Engineering: Automation and AI**

Vorlesung/4

Giese, Holger
Barkowsky, Matthias
Adriano, Christian
Ghahremani, Sona

In software engineering, like many other engineering disciplines, we on the one hand want to build solutions with the best possible quality while we on the other hand must adhere to predetermined budget and time constraints. Furthermore, often not enough qualified software engineers are available. Therefore, improving the productivity during software development and operation as well as the quality of the outcomes by automating activities partially or completely using software itself has been a major area for innovations since the early days of programming and software engineering.
Nowadays, many software engineering activities benefit from a high degree of automation and very often we take that automation for granted and are hardly aware of it anymore. Also, often the considered software systems have become so complex that they can only be developed, operated, and evolved by using largely automated approaches for various software engineering activities.
Automation in software engineering has the goal to partially or fully execute software engineering activities with minimal human intervention, thereby significantly increasing both quality and productivity. Automation successfully encompasses a broad range of activities, for instance, requirements definition, specification, software architecture, software design and synthesis, implementation, modeling, testing, quality assurance, verification, validation, maintenance, evolution, configuration management, deployment, reengineering, reuse and visualization. (…)
Also artificial intelligence is nowadays used to enhance existing software systems or make new beforehand not feasible software systems possible. Therefore, software engineering activities and outcomes have to be adjusted so that software solutions can benefit from integrated features realized with artificial intelligence. This requires that a clear strategy on how to use artificial intelligence in a software is established and that all aspects of software development and operation are appropriately adjusted to ensure that the employed combination of traditional software and artificial intelligence results in the required quality.
Therefore, we will look in this course at first into the advanced development of systems using automation for software engineering including artificial Intelligence as well as secondly into software engineering for the development of advanced systems that employ artificial intelligence. Furthermore, we will also investigate the operation of systems and how automation and in particular artificial intelligence can help there. Finally, we will discuss the case where automation and in particular artificial intelligence is used for development and operation and employed for the system itself at the same time.
We will in addition to the discussions in the lecture explore the key challenges also with small projects in the exercises and will collect at the beginning of the course suggestions for artificial intelligence tools to consider for the small projects or student presentations.
1. https://www.infoworld.com/article/3489925/github-survey-finds-nearly-all-developers-using-ai-coding-tools.html
2. https://research.google/blog/ai-in-software-engineering-at-google-progress-and-the-path-ahead/

Exam:
The grading process takes into account two components:
The results of the hands-on projects accompanying the lecture, with each project graded individually.
A final exam at the end of the semester. Depending on the number of course participants, the exam will either be oral or written.
Students will be required to pass both graded components. In particular, completing all hands-on projects to an adequate level is required for admission to the exam.
The final grade will either be composed of the average project grade (50%) and the exam grade (50%) OR the exam grade (100%) only, depending on which grading scheme yields a better result for each student individually.

036 **Software Engineering with Machine Learning: Tools and Methods**

Projektseminar/4

We will grade the group's paper report (80%) and presentations (20%). Note that the report includes documenting the experiments and the obtained results. Therefore, the grading of the report includes the experiments. During the project phase, we will require participation in meetings and other groups' presentations in the form of questions and feedback to their peers.

In the field of software engineering, the need to balance quality, budget constraints, and time limitations are constant drivers for innovation in tools and methods. Because software engineering tasks are extremely labor intensive, automation has become a critical area of focus, aiming to improve productivity during software development and operation while maintaining high-quality code and specificaitons. As a result, many software engineering tasks currently benefit from automation. Meanwhile, artificial intelligence (AI) in general and various specific Machine Learning methods have been bringing new opportunities for automation.. Even before the term “software engineering” was coined, AI was considered a candidate technology. Currently, AI is poised to revolutionize software development. Surveys show that over 97% of developers have used AI coding tools, and companies like Google already produce 50% of their code using AI. AI enhances existing software systems and enables previously unfeasible solutions. However, a clear strategy is essential to integrate AI effectively, adjusting all aspects of software development and operation to ensure the desired quality.
Finally, in this project seminar, we will develop projects that explore how to advance software engineering tasks using automation and specific machine learning methods, from Large Language Models to Reinforcement Learning and Graph Neural Networks. We will also discuss in the context of the projects the particularities of software engineering for AI-driven systems and how automation and AI impact system operation.
This project seminar is a companion of the course “Advanced Topics in Software Engineering: Automation and AI (ASE)”, in a sense that the conceptual and theoretical topics will be covered in the lecture, while the project seminar will focus on more in-depth designs and prototypes. For this reason the participants in the project seminar are invited to attend the ASE lectures.

Barkowsky, Matthias
Giese, Holger
Adriano, Christian

029 **Modeling of Embedded Systems using Graphtransformation**

Projektseminar/4

Embedded systems consist of software components that observe and control a physical environment. The discrete parts of the states of such embedded systems can be represented by graphs. The behavior of such embedded systems can then be described by various kinds of graph transformation systems capturing aspects such as time and probabilism at varying levels of detail. In this course we use tools to model and analyze embedded systems using graph transformation systems.

The course begins with an introduction to graphs, graph transformation steps, and graph transformation systems. Students will then work in groups to understand the concepts presented though manual calculation, implementation, and use of tools. Students will submit and present the results of each phase.
Phase 1: Graph transformation fundamentals.
Phase 2: Graph transformation modeling using the Groove tool, followed by analysis.
Phase 3: Graph transformation modeling using probabilistic timed graph transformation systems using the Henshin tools, followed by analysis.
Moodle Course

Exam
Modulprüfungen: Mündliche Prüfung, 30-45 Minuten
Prüfungsnebenleistungen: Für die Zulassung zur Modulprüfung: Übungsaufgaben (50%)

Giese, Holger
Maximova, Maria
Schneider, Sven

5 **Global Team-Based Innovation I**

Projektseminar/4

Uebernickel, Falk
Beermann, Vincent
Enkmann, Jan
Rolfes, Theresa Maria
Cauderay, Virginie
Wuttke, Tobias

Global Team-based Innovation (GTI) is a course designated for master students of the Hasso Plattner Institute (HPI) and the University of Potsdam (UP).
In our course, students apply IT knowledge to engineer digital solutions for real business challenges provided by prominent global companies. We follow the Design Thinking methodology to innovate on wicked problems given by our project partners. Within GTI, HPI students collaborate with students from other leading global universities: HPI is a partner in ME310 (for projects with the Stanford University) as well as part of the SUGAR Network for Design Innovation (for projects with other global universities).
https://hpi.de/uebernickel/teaching/global-team-based-innovation-gti-design-thinking.html
This class is exclusively available to students who have been accepted through our application process.

Exam
- Project work (20%)
  - Individual participation during lectures, group meetings and in project work
  - Stakeholder management
  - Project management (sticking to deadlines, etc.)
- Milestone presentations (20%)
  - GTI 1: Fall & winter presentation
  - GTI 2: Final presentation
- Tangible outcomes (20%)
  - One-Pagers for corporate partners
  - Intermediate prototypes
- Milestone documentations (40%)
  - GTI 1: Fall & winter documentation
  - GTI 2: Final documentation & videos

The estimated workload is 2-3 days per week.

Goals:
Students from Potsdam and leading global partner universities tackle design innovation challenges posed by global corporations. The 9 months (2 semesters) course focuses on the application of IT knowledge for engineering solutions to real business challenges. Further, we put emphasis on teaching students human-centered innovation methods and processes required for designers, engineers, and project managers of the future.
Within the projects, students go through an intense and iterative process of need finding, ideation, and rapid prototyping to create and evaluate new concepts. Company involvement provides the reality check necessary for teams to improve their innovation abilities. The team is supported by a professional coach, corporate liaisons, and faculty advisors.
Projects typically involve systems integration and include a mix of mechanical, electronic and software design. The results of all projects are real prototypes that have a user-centric design, are economically viable and technically feasible.

9 **Applied Probabilistic Machine Learning**

Seminar/4

Richard, Hugues
Renard, Bernhard Yves

028 **Deep Learning for Molecular Biology**

Seminar/2

Renard, Bernhard Yves
Rissom, Francesca
Heyne, Henrike
Nowicka, Melania Maria
Bartoszewicz, Jakub
Maciej

Rapid advances in both biology—through increased data availability and the insights derived from it—and in methods for handling high-dimensional data, such as deep learning architectures and computational resources, have created exciting opportunities for integrating these fields.

This seminar will examine how state-of-the-art deep learning models, including **CNNs, GNNs, Transformers,** and **Diffusion models**, are applied to **genome, RNA,** and **protein sequence** analysis.
We will explore how these advances are used to address key questions such as the effects of genetic mutations, protein structure and function prediction, and the design of new molecules for therapeutic purposes.
The course will primarily consist of **student presentations** on recent, preselected publications in these areas, followed by in-depth **discussions**.

Biological background is **not** necessary to participate in the seminar, but you will need a basic understanding of deep learning.
Good **English** skills are required to understand and discuss current literature.

In the seminar, each participant will give a presentation about a predefined topic within the research area and a short report. The final grade consists of the following parts:

- Oral presentation (60%)
- Written report (30%)
- Participation (10%)

Goals:

- Identify **current topics** and **open challenges** in the field of artificial intelligence for molecular biology
- Improve your understanding of **best practices in scientific research**
- **Effectively communicate** complex scientific topics in this field and lead a discussion
- Improving **presentation** and **writing skills**

The first three sessions will be in lecture format, providing an introduction to key biological concepts and a refresher on deep learning architectures. Following these sessions, students will give oral presentations on select scientific articles including a brief introduction to specific topics. These articles can be chosen from a list that will be presented during the initial meetings.
The seminar will be conducted on-site (with a hybrid option if needed). Please register on the course's Moodle page for further information.

Max. number of participants: 10

7 **Advanced Data Profiling**

Projektseminar/4

Data Profiling for Dynamic Data

Naumann, Felix
Kaminsky, Youri
Lindner, Daniel
Schmidl, Sebastian

https://hpi.de/naumann/teaching/current-courses/ws-24-25/advanced-data-profiling.html

Data profiling is the process of extracting metadata from datasets [1]. Researchers have proposed plenty of profiling algorithms for all different kinds of data dependencies, such as Unique Column Combinations (UCCs), Functional Dependencies (FDs), Inclusion Dependencies (INDs), or Order Dependencies (ODs), on static data in a batch process. However, many real-world datasets are constantly changing. These changes, which are inserts, updates, and deletes, also change the datasets' metadata, making it necessary to frequently re-profile the data. Unfortunately, executing the static profiling algorithms on every dataset change is excessively expense — even infeasible — because the static approaches do not leverage the knowledge about an earlier state of the dataset and its dependencies. This calls for novel incremental discovery algorithms that re-use existing profiling results to efficiently maintain data dependencies for dynamic datasets.
We will start with existing solutions to this problem for the following dependency types (depending on the number of students) and then improve upon them:

- **UCCs**: SWAN [2]
- **FDs**: DynFD [3], DHSFD [4]
- **INDs**: Shaabani's algorithm [5]
- **ODs**: list-based: IncOD [6], pointwise: IncPOD [7]

Seminar Organization
We will form teams of two students each. Every team works on one kind of data dependency. First, the teams become familiar with related work as an inspiration. Afterward, each student team develops their own ideas to profile their dependency type.
The students turn their ideas into working algorithms. There are two main goals for each algorithm:
1) The complete set of minimal or maximal dependencies must be maintained.
2) The runtime of the algorithm is to be optimized.
Datasets for benchmarking are provided to the students. Finally, the students present their approaches and write a short report.

Prior knowledge in data profiling (preferably completed Data Profiling lecture)
Good programming skills in a major programming language

013 **DQ4AI: Data Quality Assessment**

Projektseminar/4

Naumann, Felix
Ehrlinger, Lisa
Mohammed, Sedir

015 **Table Representation Learning**

Projektseminar/4

Naumann, Felix
Laskowski, Lukas
Pugnaloni, Francesco
Hoenes, Christoph

Representation learning (RL) aims to find meaningful representations of given objects to make them easier to process or understand. It finds application in various areas, e.g., cybersecurity, healthcare, time-series analysis, natural language processing, audio processing, and table understanding, and can be used to process data in different modalities, e.g., images, text, audio, or tabular data.
After the rise of foundation models, finding compact and uniform representations of different modalities of data became more important than ever, but while text and images have strong and consolidated representation methods, tabular data have been overlooked until recently. The research area that is trying to fill this gap is called table representation learning (TRL) and aims to extract meaningful information from tabular data to create expressive vectorial representations.

In this seminar, we will introduce you to the field of table representation learning, and explore together how different approaches perform in classic table-related tasks. To achieve that, we have the following plan:

Team activities: each team ideally consists of 2 students and will be assigned a specific TRL archetype, e.g., graph-based, LLM-based, word-embedding-based, etc. Your part is to choose one or more representative models from the ones proposed, implement them, and use them to solve classic table-related tasks, e.g., entity resolution, schema matching, etc.
Deliverable: The outcome of the seminar is a paper-style technical report that the teams will write collaboratively to present the results of the conducted analysis. In addition to the code, models, and datasets that have been produced.
Bonus: You will learn how to read/write a research paper and how to conduct scientific experiments and present the results in a paper.

Prerequisites:

- Python
- Basic knowledge of machine learning and deep learning

Organization
The organizational details for this seminar are as follows:

- Project seminar for master students
- Language of instruction: English
- 6 credit points, 4 SWS
- At most 6 participants (ideally, 3 teams of 2 students each)

Grading
In the seminar, each team will develop an approach and write a short report. The final grade consists of the following three parts:

- Approach (35%)
- Written report (35%)
- Midterm and final presentations (30%)

6 **Trends in Betriebssystemen und Middleware (Forschungsseminar)**

Seminar/2 *Polze, Andreas*

5 **Biostatistics & Epidemiological data analysis using R**

Vorlesung/4 *Konigorski, Stefan*

021 **Machine Learning Systems**

Projektseminar/4 *Rabl, Tilmann*
Salazar Diaz, Ricardo
Strassenburg, Nils
Tolovski, Ilin

020 **Data Processing on Modern Hardware**

Projektseminar/4 *Rabl, Tilmann*
Weisgut, Marcel

019 **Modern and Secure Internet: Design and Operations**

Vorlesung/4 *Bajpai, Vaibhav*
Ververis, Vasileios

6 **Build Your Own Programming Language**

Vorlesung/Seminar/4

Programming languages and how they work sometimes feel like magic, and the people who create those arcane technologies are often treated like wizards. In this course, students will dispel this magic and learn how to build a programming language themselves.

There will be a combined seminar/lecture every week. Every student has to continously work on the implemententation of their language and show progress every week.

- In-depth knowledge in at least one dynamic programming language
- Knowledge of Java and associated technologies helpful, but not required

Grading will take place based on the continuous work on the projects and the final oral examination. To complete the course, the following requirements are to be fulfilled, and the grade will be composed of:

- Regular submission of implementation progress (weekly) (20%)
- Functional implementation of the language at the end of the semester (30%)
- Oral exam at end of semester (50%)
- Bonus Points from weekly challenges

All source code created during this seminar will be licenced under the MIT license

Oral exam at end of semester

Hirschfeld, Robert
Lincke, Jens
Felgentreff, Tim
Niephaus, Fabio

2 **Spatial Data: Processing and Visualization Techniques**

Seminar/Praktikum/4

Richter, Rico
Wegen, Ole
Hildebrand, Justus
Schulz, Sebastian
Burmeister, Josafat-Mattias

Software Architecture and Modeling Technology (SAMT)

Konzepte und Methoden (HPI-SAMT-K)

9 **HCI Project Seminar on Virtual Reality and Personal Fabrication**

Seminar/Praktikum/4 — *Baudisch, Patrick*

3 **Creating Interactive 3D Web Apps with TypeScript**

Projektseminar/4 — *Baudisch, Patrick*

4 **Algorithmic folding**

Vorlesung/4

Baudisch, Patrick
Abdullah, Muhammad
Rambold, Lukas

035 **Advanced Topics in Software Engineering: Automation and AI**

Vorlesung/4

Giese, Holger
Barkowsky, Matthias
Adriano, Christian
Ghahremani, Sona

In software engineering, like many other engineering disciplines, we on the one hand want to build solutions with the best possible quality while we on the other hand must adhere to predetermined budget and time constraints. Furthermore, often not enough qualified software engineers are available. Therefore, improving the productivity during software development and operation as well as the quality of the outcomes by automating activities partially or completely using software itself has been a major area for innovations since the early days of programming and software engineering.
Nowadays, many software engineering activities benefit from a high degree of automation and very often we take that automation for granted and are hardly aware of it anymore. Also, often the considered software systems have become so complex that they can only be developed, operated, and evolved by using largely automated approaches for various software engineering activities.
Automation in software engineering has the goal to partially or fully execute software engineering activities with minimal human intervention, thereby significantly increasing both quality and productivity. Automation successfully encompasses a broad range of activities, for instance, requirements definition, specification, software architecture, software design and synthesis, implementation, modeling, testing, quality assurance, verification, validation, maintenance, evolution, configuration management, deployment, reengineering, reuse and visualization. (…)
Also artificial intelligence is nowadays used to enhance existing software systems or make new beforehand not feasible software systems possible. Therefore, software engineering activities and outcomes have to be adjusted so that software solutions can benefit from integrated features realized with artificial intelligence. This requires that a clear strategy on how to use artificial intelligence in a software is established and that all aspects of software development and operation are appropriately adjusted to ensure that the employed combination of traditional software and artificial intelligence results in the required quality.
Therefore, we will look in this course at first into the advanced development of systems using automation for software engineering including artificial Intelligence as well as secondly into software engineering for the development of advanced systems that employ artificial intelligence. Furthermore, we will also investigate the operation of systems and how automation and in particular artificial intelligence can help there. Finally, we will discuss the case where automation and in particular artificial intelligence is used for development and operation and employed for the system itself at the same time.
We will in addition to the discussions in the lecture explore the key challenges also with small projects in the exercises and will collect at the beginning of the course suggestions for artificial intelligence tools to consider for the small projects or student presentations.
1. https://www.infoworld.com/article/3489925/github-survey-finds-nearly-all-developers-using-ai-coding-tools.html
2. https://research.google/blog/ai-in-software-engineering-at-google-progress-and-the-path-ahead/

Exam:
The grading process takes into account two components:
The results of the hands-on projects accompanying the lecture, with each project graded individually.
A final exam at the end of the semester. Depending on the number of course participants, the exam will either be oral or written.
Students will be required to pass both graded components. In particular, completing all hands-on projects to an adequate level is required for admission to the exam.
The final grade will either be composed of the average project grade (50%) and the exam grade (50%) OR the exam grade (100%) only, depending on which grading scheme yields a better result for each student individually.

036 **Software Engineering with Machine Learning: Tools and Methods**

Projektseminar/4

We will grade the group's paper report (80%) and presentations (20%). Note that the report includes documenting the experiments and the obtained results. Therefore, the grading of the report includes the experiments. During the project phase, we will require participation in meetings and other groups' presentations in the form of questions and feedback to their peers.

In the field of software engineering, the need to balance quality, budget constraints, and time limitations are constant drivers for innovation in tools and methods. Because software engineering tasks are extremely labor intensive, automation has become a critical area of focus, aiming to improve productivity during software development and operation while maintaining high-quality code and specificaitons. As a result, many software engineering tasks currently benefit from automation. Meanwhile, artificial intelligence (AI) in general and various specific Machine Learning methods have been bringing new opportunities for automation.. Even before the term “software engineering” was coined, AI was considered a candidate technology. Currently, AI is poised to revolutionize software development. Surveys show that over 97% of developers have used AI coding tools, and companies like Google already produce 50% of their code using AI. AI enhances existing software systems and enables previously unfeasible solutions. However, a clear strategy is essential to integrate AI effectively, adjusting all aspects of software development and operation to ensure the desired quality.
Finally, in this project seminar, we will develop projects that explore how to advance software engineering tasks using automation and specific machine learning methods, from Large Language Models to Reinforcement Learning and Graph Neural Networks. We will also discuss in the context of the projects the particularities of software engineering for AI-driven systems and how automation and AI impact system operation.
This project seminar is a companion of the course “Advanced Topics in Software Engineering: Automation and AI (ASE)”, in a sense that the conceptual and theoretical topics will be covered in the lecture, while the project seminar will focus on more in-depth designs and prototypes. For this reason the participants in the project seminar are invited to attend the ASE lectures.

Barkowsky, Matthias
Giese, Holger
Adriano, Christian

029 **Modeling of Embedded Systems using Graphtransformation**

Projektseminar/4

Embedded systems consist of software components that observe and control a physical environment. The discrete parts of the states of such embedded systems can be represented by graphs. The behavior of such embedded systems can then be described by various kinds of graph transformation systems capturing aspects such as time and probabilism at varying levels of detail. In this course we use tools to model and analyze embedded systems using graph transformation systems.

The course begins with an introduction to graphs, graph transformation steps, and graph transformation systems. Students will then work in groups to understand the concepts presented though manual calculation, implementation, and use of tools. Students will submit and present the results of each phase.
Phase 1: Graph transformation fundamentals.
Phase 2: Graph transformation modeling using the Groove tool, followed by analysis.
Phase 3: Graph transformation modeling using probabilistic timed graph transformation systems using the Henshin tools, followed by analysis.
Moodle Course

Exam
Modulprüfungen: Mündliche Prüfung, 30-45 Minuten
Prüfungsnebenleistungen: Für die Zulassung zur Modulprüfung: Übungsaufgaben (50%)

Giese, Holger
Maximova, Maria
Schneider, Sven

9 **Applied Probabilistic Machine Learning**

Seminar/4

Richard, Hugues
Renard, Bernhard Yves

028 **Deep Learning for Molecular Biology**

Seminar/2 — *Renard, Bernhard Yves; Rissom, Francesca; Heyne, Henrike; Nowicka, Melania Maria; Bartoszewicz, Jakub Maciej*

Rapid advances in both biology—through increased data availability and the insights derived from it—and in methods for handling high-dimensional data, such as deep learning architectures and computational resources, have created exciting opportunities for integrating these fields.

This seminar will examine how state-of-the-art deep learning models, including **CNNs, GNNs, Transformers,** and **Diffusion models**, are applied to **genome, RNA,** and **protein sequence** analysis.
We will explore how these advances are used to address key questions such as the effects of genetic mutations, protein structure and function prediction, and the design of new molecules for therapeutic purposes.
The course will primarily consist of **student presentations** on recent, preselected publications in these areas, followed by in-depth **discussions**.

Biological background is **not** necessary to participate in the seminar, but you will need a basic understanding of deep learning.
Good **English** skills are required to understand and discuss current literature.

In the seminar, each participant will give a presentation about a predefined topic within the research area and a short report. The final grade consists of the following parts:

- Oral presentation (60%)
- Written report (30%)
- Participation (10%)

Goals:

- Identify **current topics** and **open challenges** in the field of artificial intelligence for molecular biology
- Improve your understanding of **best practices in scientific research**
- **Effectively communicate** complex scientific topics in this field and lead a discussion
- Improving **presentation** and **writing skills**

The first three sessions will be in lecture format, providing an introduction to key biological concepts and a refresher on deep learning architectures. Following these sessions, students will give oral presentations on select scientific articles including a brief introduction to specific topics. These articles can be chosen from a list that will be presented during the initial meetings.
The seminar will be conducted on-site (with a hybrid option if needed).
Please register on the course's Moodle page for further information.

Max. number of participants: 10

6 **Graphenalgorithmen**

Vorlesung/Übung/4 — *Friedrich, Tobias; Skretas, Georgios*

6 **Advanced Competitive Programming 2**

Vorlesung/4 — *Friedrich, Tobias; Simonov, Kirill; Cohen, Sarel*

6 **Trends in Betriebssystemen und Middleware (Forschungsseminar)**

Seminar/2 — *Polze, Andreas*

8 **Advanced Machine Learning Seminar**

Seminar/4 — *Lippert, Christoph*

021 **Machine Learning Systems**

Projektseminar/4 — *Rabl, Tilmann; Salazar Diaz, Ricardo; Strassenburg, Nils; Tolovski, Ilin*

020 **Data Processing on Modern Hardware**

Projektseminar/4 — *Rabl, Tilmann; Weisgut, Marcel*

6 **Build Your Own Programming Language**

Vorlesung/Seminar/4

Programming languages and how they work sometimes feel like magic, and the people who create those arcane technologies are often treated like wizards. In this course, students will dispel this magic and learn how to build a programming language themselves.

There will be a combined seminar/lecture every week. Every student has to continously work on the implemententation of their language and show progress every week.

- In-depth knowledge in at least one dynamic programming language
- Knowledge of Java and associated technologies helpful, but not required

Grading will take place based on the continuous work on the projects and the final oral examination. To complete the course, the following requirements are to be fulfilled, and the grade will be composed of:

- Regular submission of implementation progress (weekly) (20%)
- Functional implementation of the language at the end of the semester (30%)
- Oral exam at end of semester (50%)
- Bonus Points from weekly challenges

All source code created during this seminar will be licenced under the MIT license

Oral exam at end of semester

Hirschfeld, Robert
Lincke, Jens
Felgentreff, Tim
Niephaus, Fabio

Technologien und Werkzeuge (HPI-SAMT-T)

9 **HCI Project Seminar on Virtual Reality and Personal Fabrication**

Seminar/Praktikum/4 — *Baudisch, Patrick*

3 **Creating Interactive 3D Web Apps with TypeScript**

Projektseminar/4 — *Baudisch, Patrick*

4 **Algorithmic folding**

Vorlesung/4 — *Baudisch, Patrick*
Abdullah, Muhammad
Rambold, Lukas

035 **Advanced Topics in Software Engineering: Automation and AI**

Vorlesung/4

Giese, Holger
Barkowsky, Matthias
Adriano, Christian
Ghahremani, Sona

In software engineering, like many other engineering disciplines, we on the one hand want to build solutions with the best possible quality while we on the other hand must adhere to predetermined budget and time constraints. Furthermore, often not enough qualified software engineers are available. Therefore, improving the productivity during software development and operation as well as the quality of the outcomes by automating activities partially or completely using software itself has been a major area for innovations since the early days of programming and software engineering.
Nowadays, many software engineering activities benefit from a high degree of automation and very often we take that automation for granted and are hardly aware of it anymore. Also, often the considered software systems have become so complex that they can only be developed, operated, and evolved by using largely automated approaches for various software engineering activities.
Automation in software engineering has the goal to partially or fully execute software engineering activities with minimal human intervention, thereby significantly increasing both quality and productivity. Automation successfully encompasses a broad range of activities, for instance, requirements definition, specification, software architecture, software design and synthesis, implementation, modeling, testing, quality assurance, verification, validation, maintenance, evolution, configuration management, deployment, reengineering, reuse and visualization. (…)
Also artificial intelligence is nowadays used to enhance existing software systems or make new beforehand not feasible software systems possible. Therefore, software engineering activities and outcomes have to be adjusted so that software solutions can benefit from integrated features realized with artificial intelligence. This requires that a clear strategy on how to use artificial intelligence in a software is established and that all aspects of software development and operation are appropriately adjusted to ensure that the employed combination of traditional software and artificial intelligence results in the required quality.
Therefore, we will look in this course at first into the advanced development of systems using automation for software engineering including artificial Intelligence as well as secondly into software engineering for the development of advanced systems that employ artificial intelligence. Furthermore, we will also investigate the operation of systems and how automation and in particular artificial intelligence can help there. Finally, we will discuss the case where automation and in particular artificial intelligence is used for development and operation and employed for the system itself at the same time.
We will in addition to the discussions in the lecture explore the key challenges also with small projects in the exercises and will collect at the beginning of the course suggestions for artificial intelligence tools to consider for the small projects or student presentations.
1. https://www.infoworld.com/article/3489925/github-survey-finds-nearly-all-developers-using-ai-coding-tools.html
2. https://research.google/blog/ai-in-software-engineering-at-google-progress-and-the-path-ahead/

Exam:
The grading process takes into account two components:
The results of the hands-on projects accompanying the lecture, with each project graded individually.
A final exam at the end of the semester. Depending on the number of course participants, the exam will either be oral or written.
Students will be required to pass both graded components. In particular, completing all hands-on projects to an adequate level is required for admission to the exam.
The final grade will either be composed of the average project grade (50%) and the exam grade (50%) OR the exam grade (100%) only, depending on which grading scheme yields a better result for each student individually.

036 **Software Engineering with Machine Learning: Tools and Methods**

Projektseminar/4

We will grade the group's paper report (80%) and presentations (20%). Note that the report includes documenting the experiments and the obtained results. Therefore, the grading of the report includes the experiments. During the project phase, we will require participation in meetings and other groups' presentations in the form of questions and feedback to their peers.

In the field of software engineering, the need to balance quality, budget constraints, and time limitations are constant drivers for innovation in tools and methods. Because software engineering tasks are extremely labor intensive, automation has become a critical area of focus, aiming to improve productivity during software development and operation while maintaining high-quality code and specificaitons. As a result, many software engineering tasks currently benefit from automation. Meanwhile, artificial intelligence (AI) in general and various specific Machine Learning methods have been bringing new opportunities for automation.. Even before the term “software engineering” was coined, AI was considered a candidate technology. Currently, AI is poised to revolutionize software development. Surveys show that over 97% of developers have used AI coding tools, and companies like Google already produce 50% of their code using AI. AI enhances existing software systems and enables previously unfeasible solutions. However, a clear strategy is essential to integrate AI effectively, adjusting all aspects of software development and operation to ensure the desired quality.
Finally, in this project seminar, we will develop projects that explore how to advance software engineering tasks using automation and specific machine learning methods, from Large Language Models to Reinforcement Learning and Graph Neural Networks. We will also discuss in the context of the projects the particularities of software engineering for AI-driven systems and how automation and AI impact system operation.
This project seminar is a companion of the course “Advanced Topics in Software Engineering: Automation and AI (ASE)”, in a sense that the conceptual and theoretical topics will be covered in the lecture, while the project seminar will focus on more in-depth designs and prototypes. For this reason the participants in the project seminar are invited to attend the ASE lectures.

Barkowsky, Matthias
Giese, Holger
Adriano, Christian

029 **Modeling of Embedded Systems using Graphtransformation**

Projektseminar/4

Embedded systems consist of software components that observe and control a physical environment. The discrete parts of the states of such embedded systems can be represented by graphs. The behavior of such embedded systems can then be described by various kinds of graph transformation systems capturing aspects such as time and probabilism at varying levels of detail. In this course we use tools to model and analyze embedded systems using graph transformation systems.

The course begins with an introduction to graphs, graph transformation steps, and graph transformation systems. Students will then work in groups to understand the concepts presented though manual calculation, implementation, and use of tools. Students will submit and present the results of each phase.
Phase 1: Graph transformation fundamentals.
Phase 2: Graph transformation modeling using the Groove tool, followed by analysis.
Phase 3: Graph transformation modeling using probabilistic timed graph transformation systems using the Henshin tools, followed by analysis.
Moodle Course

Exam
Modulprüfungen: Mündliche Prüfung, 30-45 Minuten
Prüfungsnebenleistungen: Für die Zulassung zur Modulprüfung: Übungsaufgaben (50%)

Giese, Holger
Maximova, Maria
Schneider, Sven

9 **Applied Probabilistic Machine Learning**

Seminar/4

Richard, Hugues
Renard, Bernhard Yves

028 **Deep Learning for Molecular Biology**

Seminar/2

Renard, Bernhard Yves
Rissom, Francesca
Heyne, Henrike
Nowicka, Melania Maria
Bartoszewicz, Jakub
Maciej

Rapid advances in both biology—through increased data availability and the insights derived from it—and in methods for handling high-dimensional data, such as deep learning architectures and computational resources, have created exciting opportunities for integrating these fields.

This seminar will examine how state-of-the-art deep learning models, including **CNNs, GNNs, Transformers,** and **Diffusion models**, are applied to **genome, RNA,** and **protein sequence** analysis.
We will explore how these advances are used to address key questions such as the effects of genetic mutations, protein structure and function prediction, and the design of new molecules for therapeutic purposes.
The course will primarily consist of **student presentations** on recent, preselected publications in these areas, followed by in-depth **discussions**.

Biological background is **not** necessary to participate in the seminar, but you will need a basic understanding of deep learning.
Good **English** skills are required to understand and discuss current literature.

In the seminar, each participant will give a presentation about a predefined topic within the research area and a short report. The final grade consists of the following parts:

- Oral presentation (60%)
- Written report (30%)
- Participation (10%)

Goals:

- Identify **current topics** and **open challenges** in the field of artificial intelligence for molecular biology
- Improve your understanding of **best practices in scientific research**
- **Effectively communicate** complex scientific topics in this field and lead a discussion
- Improving **presentation** and **writing skills**

The first three sessions will be in lecture format, providing an introduction to key biological concepts and a refresher on deep learning architectures. Following these sessions, students will give oral presentations on select scientific articles including a brief introduction to specific topics. These articles can be chosen from a list that will be presented during the initial meetings.
The seminar will be conducted on-site (with a hybrid option if needed).
Please register on the course's Moodle page for further information.

Max. number of participants: 10

6 **Graphenalgorithmen**

Vorlesung/Übung/
4

Friedrich, Tobias
Skretas, Georgios

6 **Advanced Competitive Programming 2**

Vorlesung/4

Friedrich, Tobias
Simonov, Kirill
Cohen, Sarel

003 **Understanding Graphs, Algorithms, Randomness**

Seminar/2

Friedrich, Tobias
Goebel, Andreas
Verma, Shaily

005 **Advanced Topics in Algorithms and Complexity**

Vorlesung/4

Friedrich, Tobias
Goebel, Andreas
Verma, Shaily

045 **Algorithms for Collective Decision Making**

Vorlesung/Übung/4 — *Boehmer, Niclas*

This module deals with collective decision making, where a group of agents with preferences over alternatives seeks to select a compromise alternative that fairly reflects everyone's preferences. We focus on three types of collective decision making scenarios:
Voting: Selecting one or more candidates to represent a population of voters based on their preferences over candidates**.**
Resource Allocation: Fairly and efficiently distributing a set of items among agents.
Coalition Formation: Dividing agents into teams based on their preferences for different teams.
The course takes a primarily theoretical approach to these problems, rooted in computational social choice, a field at the intersection of theoretical computer science and economics. We study collective decision making problems from four perspectives, which are all also relevant beyond computational social choice:
Algorithmic: How efficiently can we find a winning alternative?
Axiomatic: Can we design an algorithm that satisfies a set of desirable normative properties?
Game-theoretic: Can agents strategically manipulate the algorithm/outcome?
Experimental: How do different algorithms behave in practice?

The course will consist of three parts: Voting, resource allocation, and coalition formation, where the first part is roughly as long as the other two combined. Covered topics include:
Voting

- Single Winner Voting & Rank Aggregation: voting rules, winner determination problem, axiomatic characterizations and impossibility results, manipulation, robustness, other computational problems around elections
- Multiwinner Voting & Participatory Budgeting: Voting rules, winner determination problem, proportionality axioms, transparency, real-world instances
- Applications: clustering, proof-of-stake blockchain, deliberation, LLMs / reinforcement learning from human feedback

Resource Allocation

- Divisible Goods: fairness axioms, Robertson-Webb model and query complexity, price of proportionality
- Indivisible Goods: fairness axioms, computing fair allocations

Coalition Formation/ Cooperative Game Theory

- Transferable utilities: stability concepts, Shapely value and its applications
- Non-transferable utilities: hedonic games and stable matching, stability concepts, computing stable outcomes

Final Exam: The planned exam mode is a ~30-minute oral exam, which will constitute 100% of the course grade. An average grade of at least 50% in the exercises is required for students to participate in the final exam but does not contribute towards the course grade.
Exercises: Exercises will be assigned on a (bi-)weekly basis and will consist of two types: (1) Traditional problem-solving exercise sheets and (2) Readings of (parts of) research papers, accompanied by comprehension questions.

6 **Trends in Betriebssystemen und Middleware (Forschungsseminar)**

Seminar/2 — *Polze, Andreas*

8 **Advanced Machine Learning Seminar**

Seminar/4 — *Lippert, Christoph*

021 **Machine Learning Systems**

Projektseminar/4 — *Rabl, Tilmann*
Salazar Diaz, Ricardo
Strassenburg, Nils
Tolovski, Ilin

020 **Data Processing on Modern Hardware**

Projektseminar/4 — *Rabl, Tilmann*
Weisgut, Marcel

6	**Build Your Own Programming Language**		
	Vorlesung/Seminar/4	Programming languages and how they work sometimes feel like magic, and the people who create those arcane technologies are often treated like wizards. In this course, students will dispel this magic and learn how to build a programming language themselves. There will be a combined seminar/lecture every week. Every student has to continously work on the implemententation of their language and show progress every week. • In-depth knowledge in at least one dynamic programming language • Knowledge of Java and associated technologies helpful, but not required Grading will take place based on the continuous work on the projects and the final oral examination. To complete the course, the following requirements are to be fulfilled, and the grade will be composed of: • Regular submission of implementation progress (weekly) (20%) • Functional implementation of the language at the end of the semester (30%) • Oral exam at end of semester (50%) • Bonus Points from weekly challenges All source code created during this seminar will be licenced under the MIT license Oral exam at end of semester	*Hirschfeld, Robert* *Lincke, Jens* *Felgentreff, Tim* *Niephaus, Fabio*

Spezialisierung (HPI-SAMT-S)

9	**HCI Project Seminar on Virtual Reality and Personal Fabrication**		
	Seminar/Praktikum/4		*Baudisch, Patrick*
3	**Creating Interactive 3D Web Apps with TypeScript**		
	Projektseminar/4		*Baudisch, Patrick*
4	**Algorithmic folding**		
	Vorlesung/4		*Baudisch, Patrick* *Abdullah, Muhammad* *Rambold, Lukas*

035 **Advanced Topics in Software Engineering: Automation and AI**

Vorlesung/4

Giese, Holger
Barkowsky, Matthias
Adriano, Christian
Ghahremani, Sona

In software engineering, like many other engineering disciplines, we on the one hand want to build solutions with the best possible quality while we on the other hand must adhere to predetermined budget and time constraints. Furthermore, often not enough qualified software engineers are available. Therefore, improving the productivity during software development and operation as well as the quality of the outcomes by automating activities partially or completely using software itself has been a major area for innovations since the early days of programming and software engineering.
Nowadays, many software engineering activities benefit from a high degree of automation and very often we take that automation for granted and are hardly aware of it anymore. Also, often the considered software systems have become so complex that they can only be developed, operated, and evolved by using largely automated approaches for various software engineering activities.
Automation in software engineering has the goal to partially or fully execute software engineering activities with minimal human intervention, thereby significantly increasing both quality and productivity. Automation successfully encompasses a broad range of activities, for instance, requirements definition, specification, software architecture, software design and synthesis, implementation, modeling, testing, quality assurance, verification, validation, maintenance, evolution, configuration management, deployment, reengineering, reuse and visualization. (…)
Also artificial intelligence is nowadays used to enhance existing software systems or make new beforehand not feasible software systems possible. Therefore, software engineering activities and outcomes have to be adjusted so that software solutions can benefit from integrated features realized with artificial intelligence. This requires that a clear strategy on how to use artificial intelligence in a software is established and that all aspects of software development and operation are appropriately adjusted to ensure that the employed combination of traditional software and artificial intelligence results in the required quality.
Therefore, we will look in this course at first into the advanced development of systems using automation for software engineering including artificial Intelligence as well as secondly into software engineering for the development of advanced systems that employ artificial intelligence. Furthermore, we will also investigate the operation of systems and how automation and in particular artificial intelligence can help there. Finally, we will discuss the case where automation and in particular artificial intelligence is used for development and operation and employed for the system itself at the same time.
We will in addition to the discussions in the lecture explore the key challenges also with small projects in the exercises and will collect at the beginning of the course suggestions for artificial intelligence tools to consider for the small projects or student presentations.
1. https://www.infoworld.com/article/3489925/github-survey-finds-nearly-all-developers-using-ai-coding-tools.html
2. https://research.google/blog/ai-in-software-engineering-at-google-progress-and-the-path-ahead/

Exam:
The grading process takes into account two components:
The results of the hands-on projects accompanying the lecture, with each project graded individually.
A final exam at the end of the semester. Depending on the number of course participants, the exam will either be oral or written.
Students will be required to pass both graded components. In particular, completing all hands-on projects to an adequate level is required for admission to the exam.
The final grade will either be composed of the average project grade (50%) and the exam grade (50%) OR the exam grade (100%) only, depending on which grading scheme yields a better result for each student individually.

036 **Software Engineering with Machine Learning: Tools and Methods**

Projektseminar/4

We will grade the group's paper report (80%) and presentations (20%). Note that the report includes documenting the experiments and the obtained results. Therefore, the grading of the report includes the experiments. During the project phase, we will require participation in meetings and other groups' presentations in the form of questions and feedback to their peers.

In the field of software engineering, the need to balance quality, budget constraints, and time limitations are constant drivers for innovation in tools and methods. Because software engineering tasks are extremely labor intensive, automation has become a critical area of focus, aiming to improve productivity during software development and operation while maintaining high-quality code and specificaitons. As a result, many software engineering tasks currently benefit from automation. Meanwhile, artificial intelligence (AI) in general and various specific Machine Learning methods have been bringing new opportunities for automation.. Even before the term “software engineering” was coined, AI was considered a candidate technology. Currently, AI is poised to revolutionize software development. Surveys show that over 97% of developers have used AI coding tools, and companies like Google already produce 50% of their code using AI. AI enhances existing software systems and enables previously unfeasible solutions. However, a clear strategy is essential to integrate AI effectively, adjusting all aspects of software development and operation to ensure the desired quality.
Finally, in this project seminar, we will develop projects that explore how to advance software engineering tasks using automation and specific machine learning methods, from Large Language Models to Reinforcement Learning and Graph Neural Networks. We will also discuss in the context of the projects the particularities of software engineering for AI-driven systems and how automation and AI impact system operation.
This project seminar is a companion of the course “Advanced Topics in Software Engineering: Automation and AI (ASE)”, in a sense that the conceptual and theoretical topics will be covered in the lecture, while the project seminar will focus on more in-depth designs and prototypes. For this reason the participants in the project seminar are invited to attend the ASE lectures.

Barkowsky, Matthias
Giese, Holger
Adriano, Christian

029 **Modeling of Embedded Systems using Graphtransformation**

Projektseminar/4

Embedded systems consist of software components that observe and control a physical environment. The discrete parts of the states of such embedded systems can be represented by graphs. The behavior of such embedded systems can then be described by various kinds of graph transformation systems capturing aspects such as time and probabilism at varying levels of detail. In this course we use tools to model and analyze embedded systems using graph transformation systems.

The course begins with an introduction to graphs, graph transformation steps, and graph transformation systems. Students will then work in groups to understand the concepts presented though manual calculation, implementation, and use of tools. Students will submit and present the results of each phase.
Phase 1: Graph transformation fundamentals.
Phase 2: Graph transformation modeling using the Groove tool, followed by analysis.
Phase 3: Graph transformation modeling using probabilistic timed graph transformation systems using the Henshin tools, followed by analysis.
Moodle Course

Exam
Modulprüfungen: Mündliche Prüfung, 30-45 Minuten
Prüfungsnebenleistungen: Für die Zulassung zur Modulprüfung: Übungsaufgaben (50%)

Giese, Holger
Maximova, Maria
Schneider, Sven

9 **Applied Probabilistic Machine Learning**

Seminar/4

Richard, Hugues
Renard, Bernhard Yves

028 **Deep Learning for Molecular Biology**

Seminar/2

Renard, Bernhard Yves
Rissom, Francesca
Heyne, Henrike
Nowicka, Melania Maria
Bartoszewicz, Jakub
Maciej

Rapid advances in both biology—through increased data availability and the insights derived from it—and in methods for handling high-dimensional data, such as deep learning architectures and computational resources, have created exciting opportunities for integrating these fields.

This seminar will examine how state-of-the-art deep learning models, including **CNNs, GNNs, Transformers,** and **Diffusion models**, are applied to **genome, RNA,** and **protein sequence** analysis.
We will explore how these advances are used to address key questions such as the effects of genetic mutations, protein structure and function prediction, and the design of new molecules for therapeutic purposes.
The course will primarily consist of **student presentations** on recent, preselected publications in these areas, followed by in-depth **discussions**.

Biological background is **not** necessary to participate in the seminar, but you will need a basic understanding of deep learning.
Good **English** skills are required to understand and discuss current literature.

In the seminar, each participant will give a presentation about a predefined topic within the research area and a short report. The final grade consists of the following parts:

- Oral presentation (60%)
- Written report (30%)
- Participation (10%)

Goals:

- Identify **current topics** and **open challenges** in the field of artificial intelligence for molecular biology
- Improve your understanding of **best practices in scientific research**
- **Effectively communicate** complex scientific topics in this field and lead a discussion
- Improving **presentation** and **writing skills**

The first three sessions will be in lecture format, providing an introduction to key biological concepts and a refresher on deep learning architectures. Following these sessions, students will give oral presentations on select scientific articles including a brief introduction to specific topics. These articles can be chosen from a list that will be presented during the initial meetings.
The seminar will be conducted on-site (with a hybrid option if needed).
Please register on the course's Moodle page for further information.

Max. number of participants: 10

003 **Understanding Graphs, Algorithms, Randomness**

Seminar/2

Friedrich, Tobias
Goebel, Andreas
Verma, Shaily

005 **Advanced Topics in Algorithms and Complexity**

Vorlesung/4

Friedrich, Tobias
Goebel, Andreas
Verma, Shaily

045 **Algorithms for Collective Decision Making**

Vorlesung/Übung/4 — *Boehmer, Niclas*

This module deals with collective decision making, where a group of agents with preferences over alternatives seeks to select a compromise alternative that fairly reflects everyone's preferences. We focus on three types of collective decision making scenarios:
Voting: Selecting one or more candidates to represent a population of voters based on their preferences over candidates**.**
Resource Allocation: Fairly and efficiently distributing a set of items among agents.
Coalition Formation: Dividing agents into teams based on their preferences for different teams.
The course takes a primarily theoretical approach to these problems, rooted in computational social choice, a field at the intersection of theoretical computer science and economics. We study collective decision making problems from four perspectives, which are all also relevant beyond computational social choice:
Algorithmic: How efficiently can we find a winning alternative?
Axiomatic: Can we design an algorithm that satisfies a set of desirable normative properties?
Game-theoretic: Can agents strategically manipulate the algorithm/outcome?
Experimental: How do different algorithms behave in practice?

The course will consist of three parts: Voting, resource allocation, and coalition formation, where the first part is roughly as long as the other two combined. Covered topics include:
Voting

- Single Winner Voting & Rank Aggregation: voting rules, winner determination problem, axiomatic characterizations and impossibility results, manipulation, robustness, other computational problems around elections
- Multiwinner Voting & Participatory Budgeting: Voting rules, winner determination problem, proportionality axioms, transparency, real-world instances
- Applications: clustering, proof-of-stake blockchain, deliberation, LLMs / reinforcement learning from human feedback

Resource Allocation

- Divisible Goods: fairness axioms, Robertson-Webb model and query complexity, price of proportionality
- Indivisible Goods: fairness axioms, computing fair allocations

Coalition Formation/ Cooperative Game Theory

- Transferable utilities: stability concepts, Shapely value and its applications
- Non-transferable utilities: hedonic games and stable matching, stability concepts, computing stable outcomes

Final Exam: The planned exam mode is a ~30-minute oral exam, which will constitute 100% of the course grade. An average grade of at least 50% in the exercises is required for students to participate in the final exam but does not contribute towards the course grade.
Exercises: Exercises will be assigned on a (bi-)weekly basis and will consist of two types: (1) Traditional problem-solving exercise sheets and (2) Readings of (parts of) research papers, accompanied by comprehension questions.

6 **Trends in Betriebssystemen und Middleware (Forschungsseminar)**

Seminar/2 — *Polze, Andreas*

8 **Advanced Machine Learning Seminar**

Seminar/4 — *Lippert, Christoph*

021 **Machine Learning Systems**

Projektseminar/4 — *Rabl, Tilmann*
Salazar Diaz, Ricardo
Strassenburg, Nils
Tolovski, Ilin

020 **Data Processing on Modern Hardware**

Projektseminar/4 — *Rabl, Tilmann*
Weisgut, Marcel

6 **Build Your Own Programming Language**

Vorlesung/Seminar/4

Programming languages and how they work sometimes feel like magic, and the people who create those arcane technologies are often treated like wizards. In this course, students will dispel this magic and learn how to build a programming language themselves.

There will be a combined seminar/lecture every week. Every student has to continously work on the implemententation of their language and show progress every week.

- In-depth knowledge in at least one dynamic programming language
- Knowledge of Java and associated technologies helpful, but not required

Grading will take place based on the continuous work on the projects and the final oral examination. To complete the course, the following requirements are to be fulfilled, and the grade will be composed of:

- Regular submission of implementation progress (weekly) (20%)
- Functional implementation of the language at the end of the semester (30%)
- Oral exam at end of semester (50%)
- Bonus Points from weekly challenges

All source code created during this seminar will be licenced under the MIT license

Oral exam at end of semester

Hirschfeld, Robert
Lincke, Jens
Felgentreff, Tim
Niephaus, Fabio

Professional Skills (SSK)

Recht und Wirtschaft (HPI-SSK-RW)

2 **Founder Fundamentals I**

Vorlesung/2

Pawlitschek, Frank
Hahn, David

042 **IT-Recht**

Vorlesung/2

Brandi-Dohrn, Anselm
Menz, Monika
Fuerstenberg, Anja

011 **Rechtsfragen des „Data Engineering“**

Blockseminar/2

Die Veranstaltung vermittelt einen Überblick über die rechtlichen Anforderungen an die Entwicklung und den Vertrieb rechtskonformer digitaler Produkte bzw. Dienste und der ihnen zugrundeliegenden digitalen Geschäftsmodelle, wobei das Zusammenwirken von Jurist*Innen und Informatiker*Innen eine besondere Rolle spielt. Ferner werden Schutzmöglichkeiten digitaler Produkte dargestellt. Abschließend werden der rechtskonforme Außenauftritt eines Unternehmens und Marketingmaßnahmen besprochen.

Die Note ergibt sich aus einer Abschlussklausur (100 %)

Vermittelte Kompetenzen:

- Prüfung der rechtlichen Herausforderungen für digitale Produkte und Dienstleistungen
- Fähigkeit zum Dialog zwischen Jurist*Innen und Informatiker*Innen

Im Rahmen der Vorlesung wird das notwendige theoretische Wissen vermittelt. Darüber hinaus werden den Studierenden auch allgemeine praktische Hilfestellungen an die Hand gegeben, damit sich die Unternehmer*innen von morgen selbstständig in für sie relevanten Rechtsbereichen zurechtfinden und befähigt werden, in der Praxis die richtigen Fragen zu stellen.

Paschke, Anne
Fuerstenberg, Anja

016 **Unternehmenssimulation Strategisches Management**

Blockseminar/2

Braun, Tobias
Dabitz, Robert
Fuerstenberg, Anja

In dieser Veranstaltung erarbeiten sich die Teilnehmer zunächst im Selbststudium die Grundlagen strategischen Managements, festigen diese Kenntnisse im Rahmen eines Fallstudienseminars (Diskussion von Praxisfällen) und transferieren sie schließlich im Rahmen einer zweitätigen interaktiven Unternehmenssimulation („Berlinsim - digitale Transformation“) in die (simulierte) Führungspraxis.

Schwerpunktthemen

Strategisches Entscheiden unter Unsicherheit, strategische Umweltanalyse, Unternehmensanalyse, Wettbewerbsstrategie (Kostenschwerpunkt, Differenzierung, Stuck-in-the-middle, Hybridposition), Gesamtunternehmensstrategie (Parenting Advantage; Portfolio-Management), Strategieimplementation, Strategische Kontrolle

Exam

Leistung in der Unternehmenssimulation (50%; Kriterien werden zu Beginn der Veranstaltung bekannt gegeben), Hausarbeit (Reflexion der eigenen Entscheidungspraxis aus der Simulation vor dem Hintergrund der Modelle und Methoden des strategischen Managements; 50%; ggf. als Gruppenhausarbeit)

Entwicklung und Verankerung eines branchenunabhängigen robusten mentalen Modells strategischer Unternehmensführung

Fallstudiendiskussion, Unternehmenssimulation (Gruppenentscheidungen, Einsatz strategischer Analysetools, Coaching), Erfahrungsbasiertes Lernen, Selbststudium.

8 **Product Builder**

Seminar/4

Pawlitschek, Frank
Hahn, David

Kommunikation (HPI-SSK-KO)

5 **Academic Writing for Science**

Seminar/2

Fuerstenberg, Anja
Nemeth, Sharon

“Scientific writing is not a science. It does not contain laws obtained through derivations and experiments. Scientific writing is a craft. It consists of skills that are developed through study and practice. Moreover, scientific writing is not mystical. In fact, scientific writing is straightforward. Unlike other forms of writing … scientific writing has two specific goals: to inform readers and to persuade readers.”

– Michael Alley, “*The Craft of Scientific Writing”*

“Things should be made as simple as possible, but not any simpler.”

– Albert Einstein

The course, “Academic Writing for Science” aims to take the mystery out of scientific writing by providing knowledge and practice in the skills necessary to produce a well-written scientific paper in English. Our focus is on those qualities crucial to the positive reception of written work within the scientific community.

Class members are required to give a short presentation based on their assessment of a writing excerpt (maximum 2 pages) from a scientific text of their choice.

Participants learn what comprises clear, concise, and effective written expression. We practice identifying and resolving problems in areas that are often obstacles to good writing. In this sense, we target language and punctuation.

In new course content, participants also learn how to structure and design sentences and paragraphs for the most effective presentation of written work. The principles we learn will help improve *all* professional and academic writing.

Performance Measurement:

In-class participation, performance, and progress. A mid-term test and a final test, based on points covered in the course and writing exercises. The final grade is based on the average of the midterm and final exams points. The oral presentation is a pre-requisite to completion of the course.

Participation in class discussions plays an important role in this course, as does holding the oral presentation and completing writing activities.

038 **Communicating Technology Successfully - Developing Communication Strategies**

Blockseminar/2

Lux, Nadine
Fuerstenberg, Anja

The seminar is designed for students in the five master's degree programs in Digital Engineering who want to communicate their research topics in a structured manner and present them successfully. The main focus is on comprehensible communication of specialized knowledge to different target groups in different media. The seminar is designed to enable participants to:

- prepare communication strategies for complex topics from science, research and development for various target groups, and communicate transfer projects successfully
- apply a methodical toolbox with simple communication and strategy tools and
- to coach and support each other in the conception and implementation of communication tasks in a collegial exchange

Day 1 - Basics of Science and Technology Communication

Input on science and technology communication; overview of typical characteristics and problem areas, good practice examples
Input & exercise: target groups and goals, formulating messages, communicating knowledge
Input: Elevator pitch training - idea pitch for group work (day 2 and day 3), input on set-up and structure
Exercise: Preparing idea pitchs for day 2 (individual and partner exercise)

Day 2 - Idea Pitch & Communication Strategies

Warm-up: speech and voice training
Idea pitch: Presentation of project ideas, selecting topics and forming teams for the elaboration of the communication strategies
Input: Elements of communication strategies, examples of communication concepts
Exercise: Stakeholder analysis for own projects and definition of communication goals and target groups (group work)
Input & exercise: Comprehensible language, formulating core messages (group work)
Input: Communication measures, instruments, and formats
Exercise: Rapid prototyping for technology communication of own projects (group work)

Day 3 – Planning of communication activities

Input: Technology communication, examples of various media channels, including digital communication, social media, audio-visual communication, press and media work
Continuation of exercise: Rapid prototyping of own projects (group work) - focus on one measure, e.g. for social media, and its implementation (communication examples)
Presentation of prototypes - communication concepts for technology communication (group work, part 1 of graded exam)
Reality check & feedback from trainer and peers
Wrapup and briefing for the written assignment

The block seminar can be taken either as a supplement to the seminar "Communicating Technology Successfully - Developing Content and Formats " or independently.

Exam:

Idea pitch, development and presentation of first ideas for communication strategies for technology communication (50%)
Written assignment (max. 12 pages), elaboration of the communication strategies for technology communication presented in the seminar (50%)

039 **Communicating technology successfully – Developing Content and Formats**

Blockseminar/2

Lux, Nadine
Fuerstenberg, Anja

The seminar is aimed at students of the five master's programs in the field of digital engineering who want to communicate their research topics in a structured way and present them successfully. The focus is on developing successful formats and comprehensible content for communication with different target groups. The seminar is designed to enable the participants to

- communicate complex topics from science, research and development in a way that is appropriate for the target group and pass on knowledge in a comprehensible way
- apply methods for format development and
- to coach each other and to support each other in communication tasks in collegial exchange during conception and implementation.

The block seminar can be taken either as a supplement to the seminar "Communicating Technology Successfully - Developing Communication Strategies " or independently.

Day 1 - Basic knowledge of format development for science and technology communication

Input on the topic of science and technology communication; overview of typical characteristics and problem areas, best and worst practice examples
Input & exercise: understanding audiences and target groups
Exercise: text formats - comprehensible language, tips and tricks for writing
Input & exercises: Trends in research communication - social media, websites, community participation & citizen science
Input & exercise: hands-on research - Visitor centers, science centers, fairs, events & co.

Day 2 - Communicating science and technologies

Input & exercises: Media and public relations
Easy listening: Audio formats, radio & podcasts
Visualizing research: Image formats, clips and documentaries
Discussing science: Interview situations and public dialogues
Input & presentation training: My (research) project in 120 seconds; input on composition and structure (individual and partner exercise)
Input & exercise: oral presentations, body language, preparing scripts; feedback from trainer and peers

Day 3 – Developing formats for digital Science and Technology Communication

Input on format development in science and technology communication
Input & exercise: Digital storytelling for the communication of own projects (group work), storyboards & conception
Presentation of format ideas (group work, part 1 of graded exam)
Reality check and feedback from trainer and peers
Wrapup and briefing for the written assignment

Exam

- Presentation “My (research) project in 120 seconds”, development and presentation of a digital (storytelling) format for own research and/or technology communication (group work) (50%)
- Written paper (max. 12 pages), elaboration of the ideas for technology communication presented in the seminar (50%)

040 **Führungskompetenz - über die harten Auswirkungen der Soft Skills**

Blockseminar/2

Heidemann, Michael Karl
Unger, Eugen
Fuerstenberg, Anja

Fachliche Kompetenzen werden in Unternehmen als selbstverständlich vorausgesetzt. Das Seminar geht von der These aus, dass mit jedem Karriereschritt in der Hierarchie auch die Anforderungen an soziale Kompetenz (Kommunikationsfähigkeit, Konfliktfähigkeit, Werteorientierung) steigen.
Modul 1 - Referent Michael Karl Heidemann
Führung in Veränderungsprozessen: Unternehmenskultur gestalten
Verantwortung in Unternehmen zu tragen, heißt heute vor allem, Veränderungsprozesse zu initiieren, zu begleiten und erfolgreich zu machen. Welche Herausforderung bedeutet das für Führungskräfte? Wodurch ist die Unternehmenskultur eines Unternehmens bestimmt? Welche Faktoren spielen grundsätzlich eine Rolle, welche sind im Alltag wirksam? Lässt sich die Führungskultur eines Unternehmens beeinflussen und wenn ja – wie? Im ersten Modul der Reihe wird eine grundsätzliche, an der Führungsverantwortung orientierte Sicht auf das Thema entfaltet.

- Was ist Unternehmenskultur?
- Welche Bedeutung hat sie für den Erfolg des Unternehmens?
- Kann man Menschen verändern?
- Kann man Unternehmen verändern?
- Kulturelle Aspekte im Change Management
- Führung als Identitätsstiftung
- Herausforderungen in Veränderungsprozessen
- Autonomie und Heteronomie im Führungsalltag

Modul 2 - Referent Eugen Unger
Führungsalltag: Führungssituationen und Führungskommunikation
Führung beruht, wie alles soziale Handeln, auf Verhaltensmustern, die weitgehend automatisch, also unbewußt ablaufen. Das eigene Handeln an selbst entwickelten Qualitätsmaßstäben zu orientieren, bedeutet demnach Bewusstsein zu schaffen. Die Teilnehmer reflektieren ihr Führungsverständnis, indem sie sich mit ihren eigenen Annahmen und daraus resultierenden Verhaltensstrategien auseinandersetzen. Auf diese Weise bietet das Format einen diskursiven Rahmen für relevante Führungsthemen des Alltags und fördert damit ein klares Rollenverständnis als Führende.

- Selbstverständnis als Führungskraft
- Rollenanforderungen zwischen Zielen und Bedürfnissen
- Anerkennung, Kritik und Potentialentwicklung
- Führungskommunikation bewußt gestalten
- Feedbacksicherheit
- Motivation und Demotivatoren
- Zusammenspiel der Führungsinstrumente

Exam: Die Leistungserfassung erfolgt im Rahmen einer mündlichen Prüfung (Kolloquium).

041 **Intrapersonelle & Interpersonelle Kompetenzen**

Blockseminar/2

Leidenfrost, Jana
Fuerstenberg, Anja

049 **Managing stakeholders – The psychology and neuroscience of successfully influencing others**

Blockseminar/2

Frank, Franziska
Fuerstenberg, Anja

This seminar focuses on influencing skills and humility to measurably increase the likelihood for getting stakeholders on board – without having to pull the outdated hierarchy card (real or borrowed).
The first two classroom days will focus on the needs of those that are to be influenced. We will look at two types of rules: those that follow from our social needs and those that stem from the automatisms of our brain. Understanding and practicing them gives participants a set of tools, which they can employ in any work or life situation. We will look at the science behind the rules, use case examples that demonstrate their effectiveness and allow time to apply the rules to own situations.
The third classroom day looks at the person of the influencer and how their humility has measurable positive effects on employees, the organisation and themselves. We will visit concepts such as psychological safety, empowerment, error management, collaboration, accountability – all of which are fostered by a humble leader. Research has defined humility in such a way that 97 percent of leaders and employees find this a desirable virtue and wish to learn the ego-free view from the balcony. Yet there are stumbling blocks on the path to humility. We will look at how these can be avoided and how the benefits of humility be reaped across any nationality, age and gender.

The course will aim at the following learning objectives:

Students familiarize themselves with both the psychology and neuroscience of influencing and learn to apply the concepts to different situations. The ability to navigate different stakeholder needs and achieve synergy with their own needs is fostered.
Students develop an understanding of the value of humility. They grasp how the concept has nothing to do with weakness, being overly modest or hiding one's light under the bushel but that it is a chosen strength for every role that they have consciously taken on. They see where they stand and learn how to strengthen humility in themselves and others.
Students receive tools, a set of influencing cards for own use as well as numerous concepts that allow them to prosper as leaders while at the same time increasing their understanding of their own patterns of reactivity.

Core themes addressed are:

Rules of influencing that stem from basic human needs and how disregarding them explain many of the negative emotions that arise in every day interactions
Rules of influencing that stem from the automatisms of our brains and how these can be utilized to get people on board
Cognitive biases and elements of individual mindsets that hinder influencing success
Humility as a trainable virtue and vital for leadership in the age of self-managing organisations, agility and New Work
Measurable benefits of humility for employees, the organisation and the humble persons themselves
Avoiding stumbling blocks and making humility habitual

Exam: Preparation of classroom sessions
Do pre-work on Qualtrics
Follow-up on classroom sessions / group presentation
Work on own situation
Interact with peer coach
Test rules of influencing and each of the four sub-elements of humility in real life
Presentation of each peer group (15 minutes)
Written documentation (minimum 3 pages)
Gewichtung der Leistungen / weighting
Group presentations (in person half a day): 50%
Individual written documentation: 50%

050 **Power and Power Misuse in Organizations**

Blockseminar/2

Drath, Karsten
Fuerstenberg, Anja

Part 1: Power in Organizations. What is it? (0.75 days)
Part 2: Destructive Leaders – Born or made? (0.75 days)
Part 3: Power Misuse in Organizations (0.75 days)
Part 4: Managing Power in Organizations (0.75 day)

Exam:
Class presentation (50%)
Written exam (50%)

Design Thinking Basic (HPI-SSK-DTB)

0 **Foundations for Design Thinking**

Projekt/Seminar/6

Foundations for Design Thinking ist ein 16-wöchiges Programm, in dem die Teilnehmer grundlegende Kenntnisse, Fähigkeiten und Fertigkeiten erwerben, um die Prinzipien des Design Thinking anzuwenden und so kreatives Selbstvertrauen aufzubauen. Während des Programms, das von April bis Juli und von Oktober bis Januar läuft, arbeitest du in verschiedenen Teams unter der Leitung unserer erfahrenen Design Thinking Coaches. Wir streben ein unterstützendes und integratives Umfeld an, das Geschlechtsidentitäten, kulturellen Hintergrund und Berufserfahrung berücksichtigt.
Das Programm gibt Einblick in verschiedene Aspekte des Design Thinking und bietet die Möglichkeit, grundlegende Werkzeuge, Methoden und Denkweisen zu erlernen, die erfolgreiche, lebenszentrierte Innovationen fördern. Du tauchst in einen experimentellen Lernansatz ein, der auf Teamarbeit basiert.
Da unser Programm auf verschiedenen Perspektiven aufbaut, suchen wir Studierende und Absolvent:innen aller Disziplinen und Fachrichtungen – von Architektur, Pädagogik, IT Systems Engineering und BWL bis hin zu Zukunftsforschung.

Foundations findet ausschließlich vor Ort an der HPI School of Design Thinking und wird im Wintersemester 2024-2025 mit 6 ECTS bewertet. Die Teilnehmeranzahl ist begrenzt auf maximal 60 Personen. Das Programm ist ein 100%iges Vor-Ort-Programm. Um das Abschlusszertifikat und ECTS-Punkte zu erhalten, ist eine regelmäßige, pünktliche und physische Teilnahme an allen Programmtagen erforderlich.

Das Programm beginnt am 20.09.2024 mit dem "Experience Day". Im Wintersemester 2024-2025 finden vom 15.10.2024 bis 28.01.2025 insgesamt 20 Programmtage (meist dienstags und freitags) vor Ort an der HPI School of Design Thinking statt. Alle Programmtage sind von 9:00 Uhr bis 17:00 Uhr. Im Februar arbeiten die Studenten an ihren Projektdokumentationen.

Englisch version:
Foundations for Design Thinking is a 16-week program where participants get the basic knowledge, skills, and capabilities to apply the principles of Design Thinking to build creative confidence. During the program, which runs from April – July and October – January you will work in different teams led by our experienced Design Thinking Coaches. We aim for a supportive and inclusive environment that considers gender identities, cultural background, and professional experience.
The program gives insight into different aspects of Design Thinking and provides the opportunity to learn basic tools, methods, and mindsets that foster successful human-centered innovations. You will dive into an experimental learning approach that is based on teamwork.

Foundations take place on site at the HPI School of Design Thinking and will be graded with 6 ECTS in the winter semester 2024-2025. The number of participants is limited to a maximum of 60 people. The program is a 100% on-site program. Regular, on-time, physical class attendance is required on all program days to be awarded Completion Certificate and ECTS points

Since our program is based on different perspectives, we are looking for students and graduates from all disciplines - from Architecture, Pedagogy, IT systems Engineering or Business to Futurology.
The program starts on 20.09.2024 with the "Experience Day". In the winter semester 2024-2025, a total of 20 program days (mostly Tuesday and Friday) will take place on site at the HPI School of Design Thinking from 15.10.2024 to 28.01.2025. All program days are from 9:00 am to 5:00 pm. In February the students are working on their project documentations.

Nicolai, Claudia
Lata, Lukasz

3	**Global Design Thinking-Workshop (D-School)**		
	Projekt/Seminar/2	Die Global Design Thinking Workshops sind ein Programm, das über die reine Einführung in Design Thinking als Prozess hinausgeht. In diesem Programm erleben die Teilnehmer:innen Design Thinking als einen lebenszentrierten Ansatz und arbeiten in verschiedenen Teams an komplexen Innovationsproblemen, unterstützt von internationalen Design Thinking-Coaches. Wir kombinieren diese Arbeit an einem konkreten Innovationsprojekt mit Reflexionen zu einem spezifischen Fokusthema. Der nächste Global Design Thinking Workshop findet im März 2025 statt Our Global Design Thinking Workshops are a education concept that goes beyond the mere introduction to Design Thinking as a process. In this program participants experience Design Thinking as a life-centered approach by dealing with complex innovation problems in diverse teams and supported by international Design Thinking coaches. We combine the work on a concrete innovation project with reflections on a specific focus topic. The next Global Design Thinking Workshop will take place in March 2025!	*Nicolai, Claudia* *Osman, Sherif Hussein Ibrahim* *Juarez Rodriguez, Maria-Jose* *Klonower, Janet*
2	**Wayfinder: Self- and Leadership Development (D-School)**		
	Projekt/Seminar/2	**Wayfinder is a newly developed program by HPI D-School that adds an essential perspective to the other program offerings in the area of Design Thinking: for self-leading and designing your own well-lived life and career.** https://hpi.de/en/school-of-design-thinking/for-students/wayfinder.html Working in innovation teams requires flexibility, agility and, above all, empathy. Empathy, and thus empathic leadership, requires skills in self-awareness and self-leadership, and shaping one's own life as well as one's own career. We believe that a structured design process can help people to develop and grow. Such a process allows them to find out what they want and how to design a satisfying and successful life. By applying and developing the methods of Design Thinking combined with fundamentals from systemic coaching and self-leadership, this program aims to learn and apply tools and techniques to improve self-awareness, recognize one's own behavioral patterns and values, reflect on and expand one's context of experience to make self-efficacy a reality in the future; building on this, to explore, prototype and test new options for a successful future. The program is based on the "Designing Your Life" Concept and has been extended and further developed by the HPI School of Design Thinking. Wayfinder has **four major focus areas**: 1. Empathy and Self-Awareness: Understanding one's own values and attitudes. 2. Exploring: Shaping career and personal life with purpose and energy. 3. Prototyping: Making good choices and exploring options. 4. Iterate: Learning forward in a strong network. Session 1: 15. November 2024 (D-School, House D) Session 2: 6. December 2024 (remote) Session 3: 10. January 2025 (remote) Session 4: 31. January 2025 (D-School, House D) The Wayfinder program is aimed at HPI students as well as participants of the Design Thinking Studios of the HPI School of Design Thinking. The course is limited to 18 participants to allow for intensive exchange and reflection in small groups.	*Schwemmle, Martin* *Thal, Klaudia* *Klonower, Janet* *Nicolai, Claudia*

Design Thinking Advanced (HPI-SSK-DTA)

0	**Design Thinking Studio: Sustainability**		
	Projekt/Seminar/6		*Nicolai, Claudia* *Grundnigg, Thomas*
7	**Design Thinking Studio: Open Innovation**		
	Projektseminar/6		*Nicolai, Claudia* *Juarez Rodriguez, Maria-Jose* *Osman, Sherif Hussein Ibrahim*

2 **Wayfinder: Self- and Leadership Development (D-School)**

Projekt/Seminar/2

Schwemmle, Martin
Thal, Klaudia
Klonower, Janet
Nicolai, Claudia

Wayfinder is a newly developed program by HPI D-School that adds an essential perspective to the other program offerings in the area of Design Thinking: for self-leading and designing your own well-lived life and career.
https://hpi.de/en/school-of-design-thinking/for-students/wayfinder.html
Working in innovation teams requires flexibility, agility and, above all, empathy. Empathy, and thus empathic leadership, requires skills in self-awareness and self-leadership, and shaping one's own life as well as one's own career. We believe that a structured design process can help people to develop and grow. Such a process allows them to find out what they want and how to design a satisfying and successful life.
By applying and developing the methods of Design Thinking combined with fundamentals from systemic coaching and self-leadership, this program aims to learn and apply tools and techniques to improve self-awareness, recognize one's own behavioral patterns and values, reflect on and expand one's context of experience to make self-efficacy a reality in the future; building on this, to explore, prototype and test new options for a successful future. The program is based on the "Designing Your Life" Concept and has been extended and further developed by the HPI School of Design Thinking.

Wayfinder has **four major focus areas**:
1. Empathy and Self-Awareness: Understanding one's own values and attitudes.
2. Exploring: Shaping career and personal life with purpose and energy.
3. Prototyping: Making good choices and exploring options.
4. Iterate: Learning forward in a strong network.

Session 1: 15. November 2024 (D-School, House D)
Session 2: 6. December 2024 (remote)
Session 3: 10. January 2025 (remote)
Session 4: 31. January 2025 (D-School, House D)

The Wayfinder program is aimed at HPI students as well as participants of the Design Thinking Studios of the HPI School of Design Thinking. The course is limited to 18 participants to allow for intensive exchange and reflection in small groups.

Management und Leitung (HPI-SSK-ML)

2 **Founder Fundamentals I**

Vorlesung/2

Pawlitschek, Frank
Hahn, David

040 **Führungskompetenz - über die harten Auswirkungen der Soft Skills**

Blockseminar/2

Heidemann, Michael Karl
Unger, Eugen
Fuerstenberg, Anja

Fachliche Kompetenzen werden in Unternehmen als selbstverständlich vorausgesetzt. Das Seminar geht von der These aus, dass mit jedem Karriereschritt in der Hierarchie auch die Anforderungen an soziale Kompetenz (Kommunikationsfähigkeit, Konfliktfähigkeit, Werteorientierung) steigen.
Modul 1 - Referent Michael Karl Heidemann
Führung in Veränderungsprozessen: Unternehmenskultur gestalten
Verantwortung in Unternehmen zu tragen, heißt heute vor allem, Veränderungsprozesse zu initiieren, zu begleiten und erfolgreich zu machen. Welche Herausforderung bedeutet das für Führungskräfte? Wodurch ist die Unternehmenskultur eines Unternehmens bestimmt? Welche Faktoren spielen grundsätzlich eine Rolle, welche sind im Alltag wirksam? Lässt sich die Führungskultur eines Unternehmens beeinflussen und wenn ja – wie? Im ersten Modul der Reihe wird eine grundsätzliche, an der Führungsverantwortung orientierte Sicht auf das Thema entfaltet.

- Was ist Unternehmenskultur?
- Welche Bedeutung hat sie für den Erfolg des Unternehmens?
- Kann man Menschen verändern?
- Kann man Unternehmen verändern?
- Kulturelle Aspekte im Change Management
- Führung als Identitätsstiftung
- Herausforderungen in Veränderungsprozessen
- Autonomie und Heteronomie im Führungsalltag

Modul 2 - Referent Eugen Unger
Führungsalltag: Führungssituationen und Führungskommunikation
Führung beruht, wie alles soziale Handeln, auf Verhaltensmustern, die weitgehend automatisch, also unbewußt ablaufen. Das eigene Handeln an selbst entwickelten Qualitätsmaßstäben zu orientieren, bedeutet demnach Bewusstsein zu schaffen. Die Teilnehmer reflektieren ihr Führungsverständnis, indem sie sich mit ihren eigenen Annahmen und daraus resultierenden Verhaltensstrategien auseinandersetzen. Auf diese Weise bietet das Format einen diskursiven Rahmen für relevante Führungsthemen des Alltags und fördert damit ein klares Rollenverständnis als Führende.

- Selbstverständnis als Führungskraft
- Rollenanforderungen zwischen Zielen und Bedürfnissen
- Anerkennung, Kritik und Potentialentwicklung
- Führungskommunikation bewußt gestalten
- Feedbacksicherheit
- Motivation und Demotivatoren
- Zusammenspiel der Führungsinstrumente

Exam: Die Leistungserfassung erfolgt im Rahmen einer mündlichen Prüfung (Kolloquium).

043 **Leading Yourself and Others in a Virtual World**

Blockseminar/2

Drath, Karsten
Fuerstenberg, Anja

1. Leading Self
- Leading Self
- How does Resilience work?
- Risk- and Protective Factors
- Victim- or Shaper mode
- Interview “Leaders Talk”
- My development plan

2. Leading Others
- Management vs. Leadership
- Six Leadership Styles by Daniel Goleman
- Self Assessment: My leadership signature
- How leaders grow
- Interview “Leaders Talk”
- My development plan

3. Leading Virtually
- Leading virtual teams
- Success factors
- Self-Assessment Leading Virtually
- Interview “Leaders Talk”
- Virtual Inspiration Challenge
- My development plan

Exam:
COURSE HOMEWORK
Due 14 days after end of course:
• Hand in individual reflection journal (structured course handout with guiding questions)
• Structured essay: “My Development Plan”
GRADING
• Reflection Journal (50%)
• My Development Plan (50%)

106 **Management Essentials**

Blockseminar/2

Kearney, Eric
Fuerstenberg, Anja

The students learn about the most important aspects of managing organizations and of managing people in organizations and how to apply this knowledge to concrete challenges.

This course offers an overview of the main topics of management. We will first cover the basics of management *of* organizations (strategic leadership) and will then turn to management *in* organizations (people management). With regard to the latter, the topics include leadership and motivation, employee satisfaction, personnel selection, training and development, and employee evaluation and compensation. Management knowledge is essential for all those who at some point wish to start their own companies or strive to occupy leadership positions in organizations.

Conveyed competencies:
- Knowledge-related competencies: strategic management; methods in management research; personnel selection; job and work design; training and development; motivation; satisfaction; leadership; personnel evaluation; personnel compensation.
- Methodological competencies; case study analysis; presentation techniques.
- Social competencies; group work and discussions.

Exam: The grade will be calculated on the basis of a group presentation (30%) and a written assignment (70%). Both the group presentation and the written assignment will focus on management aspects in organizations that the students select themselves. Further details will be provided at the beginning of the course.

049 **Managing stakeholders – The psychology and neuroscience of successfully influencing others**

Blockseminar/2 *Frank, Franziska*
Fuerstenberg, Anja

This seminar focuses on influencing skills and humility to measurably increase the likelihood for getting stakeholders on board – without having to pull the outdated hierarchy card (real or borrowed).
The first two classroom days will focus on the needs of those that are to be influenced. We will look at two types of rules: those that follow from our social needs and those that stem from the automatisms of our brain. Understanding and practicing them gives participants a set of tools, which they can employ in any work or life situation. We will look at the science behind the rules, use case examples that demonstrate their effectiveness and allow time to apply the rules to own situations.
The third classroom day looks at the person of the influencer and how their humility has measurable positive effects on employees, the organisation and themselves. We will visit concepts such as psychological safety, empowerment, error management, collaboration, accountability – all of which are fostered by a humble leader. Research has defined humility in such a way that 97 percent of leaders and employees find this a desirable virtue and wish to learn the ego-free view from the balcony. Yet there are stumbling blocks on the path to humility. We will look at how these can be avoided and how the benefits of humility be reaped across any nationality, age and gender.

The course will aim at the following learning objectives:

Students familiarize themselves with both the psychology and neuroscience of influencing and learn to apply the concepts to different situations. The ability to navigate different stakeholder needs and achieve synergy with their own needs is fostered.
Students develop an understanding of the value of humility. They grasp how the concept has nothing to do with weakness, being overly modest or hiding one's light under the bushel but that it is a chosen strength for every role that they have consciously taken on. They see where they stand and learn how to strengthen humility in themselves and others.
Students receive tools, a set of influencing cards for own use as well as numerous concepts that allow them to prosper as leaders while at the same time increasing their understanding of their own patterns of reactivity.

Core themes addressed are:

Rules of influencing that stem from basic human needs and how disregarding them explain many of the negative emotions that arise in every day interactions
Rules of influencing that stem from the automatisms of our brains and how these can be utilized to get people on board
Cognitive biases and elements of individual mindsets that hinder influencing success
Humility as a trainable virtue and vital for leadership in the age of self-managing organisations, agility and New Work
Measurable benefits of humility for employees, the organisation and the humble persons themselves
Avoiding stumbling blocks and making humility habitual

Exam: Preparation of classroom sessions
Do pre-work on Qualtrics
Follow-up on classroom sessions / group presentation
Work on own situation
Interact with peer coach
Test rules of influencing and each of the four sub-elements of humility in real life
Presentation of each peer group (15 minutes)
Written documentation (minimum 3 pages)
Gewichtung der Leistungen / weighting
Group presentations (in person half a day): 50%
Individual written documentation: 50%

050 **Power and Power Misuse in Organizations**

Blockseminar/2 *Drath, Karsten*
Fuerstenberg, Anja

Part 1: Power in Organizations. What is it? (0.75 days)
Part 2: Destructive Leaders – Born or made? (0.75 days)
Part 3: Power Misuse in Organizations (0.75 days)
Part 4: Managing Power in Organizations (0.75 day)

Exam:
Class presentation (50%)
Written exam (50%)

016 **Unternehmenssimulation Strategisches Management**

Blockseminar/2

In dieser Veranstaltung erarbeiten sich die Teilnehmer zunächst im Selbststudium die Grundlagen strategischen Managements, festigen diese Kenntnisse im Rahmen eines Fallstudienseminars (Diskussion von Praxisfällen) und transferieren sie schließlich im Rahmen einer zweitätigen interaktiven Unternehmenssimulation („Berlinsim - digitale Transformation“) in die (simulierte) Führungspraxis.
Schwerpunktthemen
Strategisches Entscheiden unter Unsicherheit, strategische Umweltanalyse, Unternehmensanalyse, Wettbewerbsstrategie (Kostenschwerpunkt, Differenzierung, Stuck-in-the-middle, Hybridposition), Gesamtunternehmensstrategie (Parenting Advantage; Portfolio-Management), Strategieimplementation, Strategische Kontrolle

Exam
Leistung in der Unternehmenssimulation (50%; Kriterien werden zu Beginn der Veranstaltung bekannt gegeben), Hausarbeit (Reflexion der eigenen Entscheidungspraxis aus der Simulation vor dem Hintergrund der Modelle und Methoden des strategischen Managements; 50%; ggf. als Gruppenhausarbeit)

Entwicklung und Verankerung eines branchenunabhängigen robusten mentalen Modells strategischer Unternehmensführung

Fallstudiendiskussion, Unternehmenssimulation (Gruppenentscheidungen, Einsatz strategischer Analysetools, Coaching), Erfahrungsbasiertes Lernen, Selbststudium.

Braun, Tobias
Dabitz, Robert
Fuerstenberg, Anja

2 **Wayfinder: Self- and Leadership Development (D-School)**

Projekt/Seminar/2

Wayfinder is a newly developed program by HPI D-School that adds an essential perspective to the other program offerings in the area of Design Thinking: for self-leading and designing your own well-lived life and career.
https://hpi.de/en/school-of-design-thinking/for-students/wayfinder.html
Working in innovation teams requires flexibility, agility and, above all, empathy. Empathy, and thus empathic leadership, requires skills in self-awareness and self-leadership, and shaping one's own life as well as one's own career. We believe that a structured design process can help people to develop and grow. Such a process allows them to find out what they want and how to design a satisfying and successful life.
By applying and developing the methods of Design Thinking combined with fundamentals from systemic coaching and self-leadership, this program aims to learn and apply tools and techniques to improve self-awareness, recognize one's own behavioral patterns and values, reflect on and expand one's context of experience to make self-efficacy a reality in the future; building on this, to explore, prototype and test new options for a successful future. The program is based on the "Designing Your Life" Concept and has been extended and further developed by the HPI School of Design Thinking.

Wayfinder has **four major focus areas**:
1. Empathy and Self-Awareness: Understanding one's own values and attitudes.
2. Exploring: Shaping career and personal life with purpose and energy.
3. Prototyping: Making good choices and exploring options.
4. Iterate: Learning forward in a strong network.

Session 1: 15. November 2024 (D-School, House D)
Session 2: 6. December 2024 (remote)
Session 3: 10. January 2025 (remote)
Session 4: 31. January 2025 (D-School, House D)

The Wayfinder program is aimed at HPI students as well as participants of the Design Thinking Studios of the HPI School of Design Thinking. The course is limited to 18 participants to allow for intensive exchange and reflection in small groups.

Schwemmle, Martin
Thal, Klaudia
Klonower, Janet
Nicolai, Claudia

8 **Product Builder**

Seminar/4

Pawlitschek, Frank
Hahn, David