HC: Human-Computer Interaction

Human–computer interaction (HCI) is concerned with designing the interaction between people and computers and the construction of interfaces to afford this.
Interaction between users and computational artefacts occurs at an interface which includes both software and hardware. Interface design impacts the software life-cycle in that it should occur early; the design and implementation of core functionality can influence the user interface – for better or worse.
Because it deals with people as well as computers, as a knowledge area HCI draws on a variety of disciplinary traditions including psychology, computer science, product design, anthropology and engineering.

HC: Human Computer Interaction (4 Core-Tier1 hours, 4 Core-Tier2 hours)

	Core-Tier1 hours	Core-Tier2 hours	Includes Electives
HCI: Foundations	**4**		**N**
HCI: Designing Interaction		**4**	**N**
HCI: Programming Interactive Systems			
HCI: User-cantered design & testing			
HCI: Design for non-Mouse interfaces			
HCI: Collaboration & communication			
HCI: Statistical Methods for HCI			
HCI: Human factors & security			
HCI: Design-oriented HCI			
HCI: Mixed, Augmented and Virtual Reality			

HC/Foundations [4 Core-Tier1 hours, 0 Core-Tier2 hours]

Motivation: For end-users, the interface *is* the system. So design in this domain must be interaction-focussed and human-centred. Students need a different repertoire of techniques to address this than is provided elsewhere in the curriculum.

Topics:

- Contexts for HCI (anything with a user interface: webpage, business applications, mobile applications, games, etc.)
- Processes for user-centered development: early focus on users, empirical testing, iterative design.
- Different measures for evaluation: utility, efficiency, learnability, user satisfaction.

- Physical capabilities that inform interaction design: colour perception, ergonomics
- Cognitive models that inform interaction design: attention, perception and recognition, movement, and memory. Gulfs of expectation and execution.
- Social models that inform interaction design: culture, communication, networks and organizations.
- Principles of good design and good designers; engineering tradeoffs
- Accessibility: interfaces for differently-abled populations (e.g blind, motion-impaired)
- Interfaces for differently-aged population groups (e.g. children, 80+)

Learning Outcomes:

Students should be able to:

1. Discuss why human-centered software development is important (knowledge)
2. Summarize the basic precepts of psychological and social interaction (knowledge)
3. Develop and use a conceptual vocabulary for analyzing human interaction with software: affordance, conceptual model, feedback, and so forth (comprehension)
4. Define a user-centered design process that explicitly recognizes that the user is not like the developer or her acquaintances (comprehension)
5. Create and conduct a simple usability test for an existing software application (application)

HC/Designing Interaction [0 Core-Tier1 hours, 4 Core-Tier2 hours]

Motivation: CS students need a minimal set of well-established methods and tools to bring to interface construction.

Topics:

- Principles of different styles of interface: e.g. command line, graphical tangible.
- Basic two-dimensional design fundamentals as applied to the visual interface, including use of grid, typography, color and contrast, scale, ordering and hierarchy.)
- Task analysis
- Paper prototyping
- Basic statistics and techniques for controlled experimentation (especially in regard to web data)
- KLM evaluation
- Help & documentation
- Handling human/system failure
- User interface standards

Learning Outcomes

Students should be able to apply the principles of HCI foundations to:

1. Create a simple application, together with help & documentation, that supports a user interface (application)
2. Conduct a quantitative evaluation and discuss/report the results (application)
3. Discuss at least one national or international user interface design standard (comprehension)

HC/Programming Interactive Systems [elective]

Motivation: To take a user-experience-centred view of software development and then cover approaches and technologies to make that happen.

Topics:

- Software Architecture Patterns: Model-View controller; command objects, online, offline, [*cross reference to Software Engineering*]
- Interaction Design Patterns: visual hierarchy, navigational distance

- Event management and user interaction
- Geometry management [*cross reference to Graphics & Visual Computing*]
- Choosing interaction styles and interaction techniques
- Presenting information: navigation, representation, manipulation
- Interface animation techniques (scene graphs, etc)
- Widget classes and libraries
- Modern GUI libraries (iOS, Android, JavaFX) GUI builders and UI programming environments [*cross reference to Graphics & Visual Computing; cross reference to Platform Based Development*]
- Declarative Interface Specification: Stylesheets and DOMs
- Data-driven applications (database-backed web pages)
- Cross-platform design
- Design for resource-constrained devices (e.g. small, mobile devices)

Learning Outcomes

Students should be able to apply the principles of HCI foundations to:

1. Understand there are common approaches to design problems, and be able to explain the importance of MVC to GUI programming (knowledge)
2. Create an application with a modern l user interface (application)
3. Identify commonalities and differences in UIs across different platforms (application)
4. Explain and use GUI programming concepts: event handling, constraint-based layout management, etc (evaluation)

HC/User-centered design and testing [elective]

Motivation: An exploration of techniques to ensure that end-users are fully considered at all stages of the design process, from inception to implementation.

Topics:

- Approaches and characteristics of design process
- Functionality and usability requirements [*cross reference to Software Engineering*]
- Techniques for gathering requirements: interviews, surveys, ethnographic & contextual enquiry [*cross reference to Software Engineering*]
- Techniques and tools for analysis & presentation of requirements: reports, personas
- Prototyping techniques and tools: sketching, storyboards, low-fidelity prototyping, wireframes
- Evaluation without users, using both qualitative and quantitative techniques: walkthroughs, GOMS, expert-based analysis, heuristics, guidelines, and standards
- Evaluation with users: observation, think-aloud, interview, survey, experiment.
- Challenges to effective evaluation: sampling, generalization.
- Reporting the results of evaluations
- Internationalisation, designing for users from other cultures, cross-cultural evaluation [*cross reference to Software Engineering*]

Learning Outcomes

Students should be able to apply the principles of HCI foundations to:

1. Understand how user-centred design complements other software process models (knowledge)
2. Choose appropriate methods to support the development of a specific UI (application)
3. Use a variety of techniques to evaluate a given UI (application)
4. Use lo-fi prototyping techniques to gather, and report, user responses (application)
5. Describe the constraints and benefits of different evaluative methods (comprehension)

HC/Design for non-mouse interfaces [elective]

Motivation: As technologies evolve, new interaction styles are made possible. This knowledge unit should be considered extensible, to track emergent technology.

Topics:

- Choosing interaction styles and interaction techniques
- Representing information to users: navigation, representation, manipulation
- Approaches to design, implementation and evaluation of non-mouse interaction
    - Touch and multi-touch interfaces
    - New Windows (iPhone, Android)
    - [Speech recognition and natural language processing – *cross reference to Intelligent Systems*]
    - Wearable and tangible interfaces
    - Persuasive interaction and emotion
    - Ubiquitious and context-aware (Ubicomp)
    - Bayesian inference (e.g. predictive text, guided pointing)
    - Ambient/peripheral display and interaction

Learning Outcomes

Students should be able to apply the principles of HCI foundations to:

1. Describe when non-mouse interfaces are appropriate (knowledge)
2. Discuss the advantages (and disadvantages) of non-mouse interfaces (application)
3. Understand the interaction possibilities beyond mouse-and-pointer interfaces (comprehension)

HC/Collaboration and communication [elective]

Motivation: Computer interfaces not only support users in achieving their individual goals but also in their interaction with others, whether that is task-focussed (work or gaming) or task-unfocussed (social networking).

Topics:

- Asynchronous group communication: e-mail, forums, facebook
- Synchronous group communication: chat rooms, conferencing, online games
- Online communities
- Software characters and intelligent agents, virtual worlds and avatars [*cross referenced to agents in Intelligent Systems*]
- Social psychology
- Social networking
- Social computing

Learning Outcomes:

Students should be able to apply the principles of HCI foundations to:

1. Describe the difference between synchronous and asynchronous communication (knowledge)
2. Compare the HCI issues in individual interaction with group interaction (comprehension)
3. Discuss several issues of social concern raised by collaborative software (comprehension)
4. Discuss the HCI issues in software that embodies human intention (comprehension)

HC/Statistical methods for HCI [elective]

Motivation: Much HCI work depends on the proper use, understanding and application of statistics. This knowledge is often held by students who join the field from psychology, but less common in students with a CS background.

Topics:

- t-tests
- ANOVA
- randomization (non-parametric) testing, within v. between-subjects design
- calculating effect size
- exploratory data analysis
- presenting statistical data
- using statistical data
- using qualitative and quantitative results together

Learning Outcomes:

Students should be able to apply the principles of HCI foundations to:

1. Explain basic statistical concepts and their areas of application (knowledge)
2. Extract and articulate the statistical arguments used in papers which report HCI results (comprehension)
3. Devise appropriate statistical tests for a given HCI problem (application)

HC/Human factors and security [elective]

Motivation: Effective interface design requires basic knowledge of security psychology. Many attacks do not have a technological basis, but exploit human propensities and vulnerabilities. "Only amateurs attack machines; professionals target people" (Bruce Schneier)

Topics:

- Applied psychology and security policies
- Security economics
- Regulatory environments – responsibility, liability and self-determination
- Organizational vulnerabilities and threats
- Usability design and security [*cross reference to Security & Information Assurance*]
- Pretext, impersonation and fraud. Phishing and spear phishing [*cross reference to Security & Information Assurance*]
- Trust, privacy and deception
- Biometric authentication (camera, voice)
- Identity management

Learning Outcomes:

Students should be able to apply the principles of HCI foundations to:

4. Explain the concepts of phishing and spear phishing, and how to recognize them (knowledge)
5. Explain the concept of identity management and its importance (knowledge)
6. Describe the issues of trust in interface design with an example of a high and low trust system (knowledge)
7. Design a user interface for a security mechanism (application)

8. Analyze a security policy and/or procedures to show where they consider, or fail to consider, human factors (comprehension)

HC/Design-oriented HCI [elective]

Motivation: Some curricula will want to emphasise an understanding of the norms and values of HCI work itself as emerging from, and deployed within specific historical, disciplinary and cultural contexts.

Topics:

- Intellectual styles and perspectives to technology and its interfaces
- Consideration of HCI as a design discipline:
    - Sketching
    - Participatory design
- Critically reflective HCI
    - Critical technical practice
    - Technologies for political activism
    - Philosophy of user experience
    - Ethnography and ethnomethodology
- Indicative domains of application
    - Sustainability
    - Arts-informed computing

Learning Objectives

Students should be able to apply the principles of HCI foundations to:

1. Detail the processes of design appropriate to specific design orientations (knowledge)
2. Apply a variety of design methods to a given problem (application)
3. Understand HCI as a design-oriented discipline. (comprehension)

Mixed, Augmented and Virtual Reality [elective]

Motivation: A detailed consideration of the interface components required for the creation and development of immersive environments, especially games.

Topics:

- Output
    - Sound
    - Stereoscopic display
    - Force feedback simulation, haptic devices
- User input
    - Viewer and object tracking
    - Pose and gesture recognition
    - Accelerometers
    - Fiducial markers
    - User interface issues
- Physical modelling and rendering
    - Physical simulation: collision detection & response, animation
    - Visibility computation

    - Time-critical rendering, multiple levels of details (LOD)
- System architectures
    - Game engines
    - Mobile augmented reality
    - Flight simulators
    - CAVEs
    - Medical imaging
- Networking
    - p2p, client-server, dead reckoning, encryption, synchronization
    - Distributed collaboration

Learning Objectives:

1. Describe the optical model realized by a computer graphics system to synthesize stereoscopic view (knowledge)
2. Describe the principles of different viewer tracking technologies. (knowledge)
3. Describe the differences between geometry- and image-based virtual reality.(knowledge)
4. Describe the issues of user action synchronization and data consistency in a networked environment.(knowledge)
5. Determine the basic requirements on interface, hardware, and software configurations of a VR system for a specified application. (application)
6. To be aware of the range of possibilities for games engines, including their potential and their limitations. (comprehension)