Pannon Egyetem

Műszaki Informatikai Kar

Distance Learning MSc programme in Computer Science Engineering

Study location	Veszprém, Hungary
Study language	English
Type	Master's, distance learning
Head of the programme	Dr. Magyar Attila

Veszprém, 15th March, 2026

Pedagogical Framework of the Distance Learning MSc Programme in Computer Science Engineering

Contents

Introduction

The MSc programme in Computer Science Engineering offered by the University of Pannonia is designed as an **English-language distance learning programme** aimed primarily at **international students** seeking advanced education in Computer Science Engineering while maintaining flexibility in their learning schedule.
The programme is aligned with the university's **digital education strategy**, which aims to expand access to high-quality higher education through modern online learning environments and pedagogical approaches. By combining asynchronous online learning with periodic on-campus intensive sessions, the programme enables students to participate in a rigorous master's level education regardless of their geographical location.
The curriculum focuses on advanced topics in **Computer Science Engineering**, with particular emphasis on areas such as **embedded systems, control engineering, and modern software and computing technologies**. The programme also reflects the research strengths of the university and integrates elements of **applied research and project-based development**, enabling students to gain both theoretical knowledge and practical engineering competences.
The educational model of the programme is designed to support independent learning while ensuring continuous academic guidance. Students engage with structured online learning materials, practical assignments, and project-based tasks delivered through a digital learning environment. Periodic on-campus sessions provide opportunities for direct interaction with instructors, supervised assessments, and collaborative project work.
Throughout the programme, students gradually transition from structured coursework to increasingly independent project-based activities, culminating in a **capstone project and master's thesis** conducted under academic supervision. This structure ensures that graduates develop the analytical, technical, and problem-solving competences required for advanced professional work or research in Computer Science Engineering.
The following sections of this document present the **pedagogical model, instructional design principles, programme structure, student support mechanisms, assessment methods, and digital infrastructure** that together define the learning environment of the distance learning programme.

Pedagogical Model of the Programme

The MSc programme in Computer Science Engineering is designed as a **blended distance learning programme** that combines asynchronous online learning with limited but strategically placed on-campus intensive periods. The pedagogical model aims to support flexible learning while maintaining high academic standards and strong practical competence development. The programme integrates **self-paced learning modules, project-based learning, and research-oriented mentoring**, enabling students to develop both theoretical knowledge and professional engineering skills.

Asynchronous self-paced learning

The majority of learning activities in the programme are conducted in an **asynchronous online learning environment**, which accounts for approximately 80% of the study workload. The learning process is organised around modular course structures hosted in the institutional Moodle learning management system.
A typical learning cycle within courses follows a structured sequence:
video lecture or learning material → self-assessment quiz → practice task → project-oriented assignment
Instructional materials are typically delivered through short lecture recordings, presentation materials, and interactive learning objects. These are complemented by automated quizzes and programming assignments that allow students to immediately apply newly acquired knowledge. This structure encourages active engagement with the material and supports self-regulated learning.
Learning analytics and completion tracking features of the learning management system are used to monitor student progress. Automated notifications and activity completion indicators help students manage their learning pace and provide early feedback on their progress.

Blended learning structure and campus weeks

Although the programme primarily operates in distance learning mode, it incorporates **two intensive on-campus periods each semester** to strengthen interaction, provide hands-on learning opportunities, and conduct certain forms of assessment.
During these campus weeks students participate in activities such as:

- individual consultations with instructors and thesis supervisors
- project work and collaborative development activities
- laboratory sessions where applicable
- in-person midterm assessments or examinations

The campus periods serve as an important complement to the online learning environment by enabling direct academic interaction, strengthening the learning community, and supporting complex project-based tasks that benefit from face-to-face collaboration.

Project-based learning

Project-based learning constitutes a central element of the pedagogical approach of the programme. Practical engineering competence is developed through a sequence of progressively complex project activities distributed across the four semesters of the programme.
The project structure is organised as follows:

- **Semester 1 – Project Laboratory:** a team-based project course in which students work collaboratively on a development task. Peer-review mechanisms may be used to support collaborative evaluation and reflection.
- **Semester 2 – Individual Laboratory Project:** students complete an individual project under instructor supervision, focusing on the application of theoretical knowledge to a practical problem.
- **Semesters 3–4 – Capstone Project and Master's Thesis:** students work under the guidance of an academic supervisor to conduct an independent engineering or research-oriented project culminating in the master's thesis.

This progressive structure allows students to gradually transition from guided teamwork to independent research and development work.

Mentoring and instructor roles

The programme employs a dual instructor role model. In theoretical and technical courses instructors primarily act as **lecturers**, providing structured learning materials and guiding students through the course content. In project-oriented and thesis-related courses instructors assume the role of **mentors or supervisors**, supporting students through individual consultations and project guidance.
Consultations typically take place through the MS Teams collaboration platform or during the campus weeks. Additional online office hours may also be organised when necessary.
This mentoring structure ensures that students receive individualized academic support during complex project and thesis work.

Assessment strategy

The programme uses a **balanced assessment model** that evaluates both theoretical understanding and practical competence. The typical distribution of assessment components is:

- 30% continuous assessment through quizzes and smaller assignments
- 30% written or oral examinations assessing theoretical knowledge
- 40% project-based evaluation measuring practical implementation skills

This structure ensures that learning outcomes are assessed through multiple complementary methods and encourages continuous engagement with course materials throughout the semester.

Digital learning environment

The primary digital learning platform of the programme is the **Moodle learning management system**, which hosts course materials, assignments, quizzes, and learning progress indicators. Moodle's completion tracking and analytics tools support both students and instructors in monitoring learning progress.
Communication and consultation are supported through **Microsoft Teams**, which provides course communication channels, online meetings, and project consultation opportunities.
Together these platforms form an integrated digital learning ecosystem that supports both asynchronous learning and synchronous interaction when required.

Microlearning and modular content design

The programme aims to apply **microlearning principles** in the design of instructional materials. Course content is divided into smaller learning units that typically consist of short lecture recordings or modular learning resources accompanied by immediate practice opportunities.
Although some materials originate from traditional lecture recordings, ongoing course development aims to increase the proportion of shorter, focused learning units that support flexible learning and repeated review of complex topics.

Summary of the pedagogical approach

Overall, the programme follows a pedagogical model that combines

- asynchronous self-paced learning
- blended learning with periodic campus sessions
- project-based competence development

- research-oriented mentoring
- continuous and diversified assessment

This integrated approach supports flexible participation for students while ensuring that graduates acquire both the theoretical foundations and the practical engineering competences required for advanced professional work in Computer Science Engineering.

Programme-level Learning Outcomes Alignment

The pedagogical design of the MSc programme in Computer Science Engineering is closely aligned with the programme-level learning outcomes defined in the curriculum. The instructional methods, learning activities, and assessment strategies of the distance learning environment are designed to support the development of both **advanced theoretical knowledge and professional engineering competences** expected from graduates of the programme.
According to the curriculum, the programme aims to develop a combination of **knowledge-based competencies and professional capabilities** in the field of Computer Science Engineering. These competencies include the ability to understand advanced engineering principles, apply problem-solving techniques in information technology systems, and design and develop complex computing solutions.

Development of theoretical knowledge

Programme-level knowledge outcomes focus on the acquisition of advanced theoretical foundations in Computer Science Engineering, including scientific principles, engineering methods, and specialized knowledge in selected areas of the discipline.
These outcomes are primarily supported through:

- structured online learning materials and lecture recordings
- advanced theoretical courses in areas such as algorithms, artificial intelligence, signal processing, and embedded systems
- self-assessment quizzes and theoretical examinations

The asynchronous learning environment enables students to engage with complex theoretical material at their own pace while reinforcing understanding through practice exercises and automated feedback.

Development of engineering problem-solving capabilities

The programme places strong emphasis on the ability to apply engineering methods and analytical thinking to complex technological problems. These capabilities include identifying relationships between system components, analysing engineering problems, and designing appropriate technical solutions.
These competences are developed through:

- programming and computational assignments
- modelling and simulation tasks
- engineering design exercises within technical courses
- practical implementation tasks using contemporary software tools and development environments

The integration of practice-oriented tasks within online courses ensures that students can continuously apply theoretical concepts in realistic engineering contexts.

Development of specialised professional competences

The curriculum allows students to develop deeper knowledge in specific areas of Computer Science Engineering, such as embedded systems, intelligent systems, communication networks, or advanced computing technologies. These specialised competences are developed through advanced elective courses and project-oriented learning activities.
Students engage with complex technological topics and learn to apply specialised tools and methods relevant to modern computing systems.

Research and innovation competences

An important objective of the programme is to prepare students for research and development activities as well as for further doctoral studies.
Research-related competences are developed through:

- research methodology courses
- project-based learning activities
- independent project work
- the master's thesis

Students learn how to formulate engineering problems, design development or research solutions, analyse results, and communicate their findings in a professional manner.

Integration of competences through project-based learning

Project-based learning plays a central role in aligning the pedagogical model with programme-level learning outcomes. The sequence of project-oriented courses allows students to gradually integrate theoretical knowledge, technical skills, and research competences.

The progression from team-based projects to independent thesis work ensures that students develop the ability to:

- design complex engineering systems
- implement and evaluate technical solutions
- conduct independent research and development activities

Alignment between pedagogy and learning outcomes

Overall, the distance learning pedagogical model supports programme-level learning outcomes through the combination of:

- structured asynchronous theoretical learning
- practice-oriented assignments and exercises
- project-based competence development
- research-oriented supervision and thesis work

This alignment ensures that graduates of the programme possess both the theoretical knowledge and practical engineering skills required for advanced professional practice in Computer Science Engineering or for further studies in doctoral programmes.

Learning Design Principles of Online Courses

The online courses of the MSc programme in Computer Science Engineering are designed according to a consistent set of pedagogical and instructional design principles that support effective learning in a distance education environment. The course design framework ensures that students can progress through the learning materials independently while maintaining continuous engagement and receiving regular feedback on their learning progress.

Modular course structure

All courses are organised into clearly defined **modules or learning units**, each focusing on a specific topic or learning objective. The modular structure allows

students to progress through the material step-by-step and supports the gradual development of knowledge and skills.
Modules typically contain the following elements:

- short lecture videos or learning materials
- supporting reading materials
- self-assessment quizzes
- practice tasks or programming exercises
- project-related activities where applicable

The modular organization also enables the use of completion tracking within the learning management system, allowing students to monitor their progress and identify remaining tasks.

Structured learning sequence

Each module follows a consistent learning sequence that supports active learning and knowledge reinforcement. The typical learning cycle is structured as follows: learning material or lecture video → self-assessment quiz → practice exercise → project-related task
This structure allows students to first acquire theoretical knowledge and then immediately apply it through exercises and practical tasks.
Self-assessment quizzes provide immediate feedback and help students evaluate their understanding before progressing to more complex assignments.

Microlearning-oriented content design

Course materials are designed according to **microlearning principles**, where possible. Learning content is divided into smaller units that can be studied independently and revisited easily when needed.
Short lecture recordings and focused learning segments help students maintain concentration and support flexible study schedules, which is particularly important in distance learning contexts where students often combine their studies with professional activities.
Although some materials originate from traditional lecture recordings, course development efforts continuously aim to increase the proportion of shorter and more focused learning units.

Practice-oriented learning activities

A strong emphasis is placed on **practice-oriented learning activities**. Many courses include programming exercises, computational tasks, or problem-solving assignments that allow students to apply theoretical concepts in practical contexts.
Where applicable, automated assessment systems (such as programming graders or automatically evaluated quizzes) provide immediate feedback to students and support scalable evaluation in the online learning environment.
In addition to smaller exercises, several courses include project-based assignments that require students to integrate knowledge from multiple modules.

Self-regulated learning support

Since the programme relies heavily on asynchronous learning, the courses are designed to support **self-regulated learning**. Students are guided through the learning process by:

- clearly defined module structures
- recommended study schedules
- task deadlines
- completion tracking indicators
- automated feedback from quizzes and exercises

These features help students manage their learning progress and avoid accumulation of unfinished tasks.

Integration with the digital learning environment

All courses are hosted in the institutional **Moodle learning management system**, which serves as the central platform for learning materials, assignments, quizzes, and progress tracking.
The system supports several pedagogical functions, including:

- activity completion tracking
- automated quizzes and assignments
- submission and evaluation of coursework
- communication through course forums
- integration with external tools used in technical courses

Communication and consultations are supported through **Microsoft Teams**, which enables online meetings, project consultations, and course-related discussions when needed.

Continuous improvement of online materials

Online course materials are regularly updated based on student feedback, instructor experience, and developments in the field of Computer Science Engineering. Continuous development efforts aim to improve the clarity of instructional materials, increase the use of interactive elements, and further support active learning in the online environment.

Structure of the Distance Learning Programme

The MSc programme in Computer Science Engineering is organised as a **distance learning programme complemented by periodic on-campus intensive sessions**. The structure of the programme is designed to support flexible learning while maintaining continuous academic progression and providing opportunities for direct interaction between students and instructors.

The programme follows a four-semester structure corresponding to a total workload of **120 ECTS credits**. Each semester combines asynchronous online learning activities with short on-campus intensive periods that support consultations, collaborative work, and certain assessment activities.

Semester structure

The primary mode of instruction throughout the semester is **asynchronous online learning**. Students study course materials independently using the digital learning environment and complete assignments, quizzes, and project tasks according to the requirements of each course.

Each semester includes three main phases:

1. **Online learning period**

During most of the semester students participate in asynchronous learning activities, which include:

- studying lecture recordings and learning materials
- completing self-assessment quizzes and programming exercises
- working on assignments and project tasks
- participating in online consultations when necessary

Online learning allows students to progress through the course materials at their own pace while meeting assignment deadlines defined within each course.

2. **On-campus intensive period during the study period**

Each semester includes a **mid-semester campus week**, during which students visit the university and participate in activities that benefit from direct interaction.
Typical activities during this period include:

- personal consultations with instructors and supervisors
- collaborative project work
- laboratory sessions where applicable
- consultations related to project-based courses
- in-person midterm assessments or tests in courses where this is required

This campus week strengthens the connection between students and instructors and supports complex learning activities that are difficult to organise in a purely online environment.

3. **On-campus intensive period during the examination session**

In addition to the mid-semester campus week, students participate in a **second on-campus period during the examination session**.
During this week students complete assessment activities that require personal presence, such as:

- written examinations
- oral examinations
- project presentations and reports
- consultations related to project courses or thesis progress

This structure allows the programme to maintain the flexibility of distance learning while ensuring that key evaluation activities take place under supervised academic conditions.

Typical semester timeline
A typical semester of the programme follows the structure below:

Period	Main Activities
Weeks 1–6	Online asynchronous learning, assignments, quizzes

Week 7	Campus week (consultations, laboratory activities, midterm assessments)
Weeks 8–13	Continued online learning, project work
Examination period	Campus week (examinations, project presentations, consultations)

This structure supports a balanced combination of flexible online learning and direct academic interaction.

Types of courses in the programme
The programme includes several types of courses that differ in their learning activities and instructional approach.

Theoretical and technical courses
Theoretical and technical courses focus on the acquisition of advanced knowledge in Computer Science Engineering and related areas. These courses primarily use asynchronous learning materials combined with automated quizzes, programming exercises, and examinations.
Students study the learning materials independently and complete tasks designed to reinforce theoretical understanding and practical skills.

Project-based courses
Project-based courses emphasise the application of knowledge through practical development tasks. These courses require students to design and implement solutions to complex engineering problems.
The programme includes a sequence of project-oriented courses that gradually increase the level of independence required from students:

- **Project Laboratory (Semester 1)** – team-based development project involving collaborative work and peer evaluation
- **Individual Laboratory Project (Semester 2)** – individual project under instructor supervision
- **Diploma Laboratory (Semester 3)** – preparation and initial development of the master's thesis project
- **Master's Thesis (Semester 4)** – independent project conducted under the guidance of a supervisor

This progressive project structure allows students to develop both teamwork skills and independent research and development competences.

Supervision and consultation

Consultation and supervision are essential components of the programme structure, particularly in project-based courses and thesis work.
Students may consult with instructors and supervisors through:

- online meetings via MS Teams
- scheduled consultations during the campus weeks
- additional online office hours when necessary

Project and thesis supervision typically takes place through regular meetings agreed upon between the student and the supervisor.

Workload distribution

The programme follows the European Credit Transfer and Accumulation System (ECTS), where **one credit corresponds to approximately 30 hours of student workload**.
Student workload includes:

- studying learning materials and lecture recordings
- completing quizzes and assignments
- programming and development tasks
- project work
- preparation for examinations
- participation in consultations and campus week activities

This workload model ensures that learning activities are distributed throughout the semester and aligned with the expected learning outcomes of each course.

Digital learning environment

The digital infrastructure supporting the programme plays a central role in the organization of distance learning activities.
The **Moodle learning management system** serves as the primary platform for:

- hosting learning materials
- distributing assignments and quizzes
- tracking activity completion and student progress

- submitting coursework and receiving feedback

The **Microsoft Teams platform** is used to support communication, consultations, and online meetings. Course-related communication typically takes place within course-specific Teams channels.
Together these platforms provide an integrated digital learning environment that supports both asynchronous learning and interactive academic collaboration.

Learning progression throughout the programme

The structure of the programme supports gradual development of both theoretical knowledge and professional engineering competences.
During the first two semesters students primarily focus on acquiring advanced theoretical knowledge and applying it through guided exercises and smaller projects.
In the later semesters the emphasis shifts toward independent project work and thesis development under academic supervision.
This progression ensures that by the end of the programme students are capable of independently solving complex engineering problems and conducting project-based development or research activities in the field of Computer Science Engineering.

Role of Campus weeks

In addition to supporting consultations and assessments, the campus weeks play an important role in strengthening the academic and professional community of the programme. Distance learning environments can sometimes limit informal interaction between students and instructors; therefore, the intensive on-campus periods provide opportunities for direct professional discussion, collaborative problem solving, and networking among students. These interactions contribute to the development of teamwork and communication skills and help students integrate more effectively into the academic culture of the programme.

Student Support and Tutoring System

The MSc programme in Computer Science Engineering incorporates a structured student support system designed to assist students throughout their studies in the distance learning environment. The programme combines digital learning tools, instructor consultations, and project supervision to ensure that students receive continuous academic guidance and support.

Instructor support in courses

In theoretical and technical courses instructors primarily provide support through the digital learning environment. Students can contact instructors via the communication tools integrated into the learning platforms, including course forums in Moodle and course communication channels in Microsoft Teams.
Instructors typically respond to student inquiries within a reasonable time frame, and additional consultations may be organised when necessary. These consultations may take place online through MS Teams or during the on-campus intensive periods. Online consultations provide opportunities for students to clarify course material, discuss assignments, and receive guidance on problem-solving approaches.

Project supervision

Project-based courses and thesis-related activities include a structured supervision process. Each student or student group working on project-based assignments is supported by an instructor acting as a supervisor.
Supervision activities include:

- consultation meetings with the supervisor
- discussion of project requirements and design decisions
- feedback on project progress and deliverables
- guidance on technical and methodological issues

Consultations are typically arranged individually between the student and the supervisor and are conducted either online via MS Teams or in person the during campus weeks.

Support during thesis work

In the final phase of the programme students work on their master's thesis under the supervision of an academic advisor. The advisor provides continuous guidance throughout the thesis preparation process.
Supervision activities may include:

- defining the research or development topic
- discussing the project plan and milestones
- reviewing intermediate results and drafts
- supporting methodological decisions

Regular consultations ensure that students receive timely feedback and maintain steady progress toward the completion of the thesis.

Digital communication channels

Communication between students and instructors is supported through an integrated digital environment.
The main communication channels include:

- **Moodle course forums**, used for course-related announcements and discussions
- **Microsoft Teams channels**, used for direct communication, consultations, and project discussions
- **email communication**, used for individual inquiries when necessary

These communication channels ensure that students can easily reach instructors and receive support when needed.

Monitoring student progress

The programme uses several mechanisms to monitor student progress and identify potential difficulties early.
The Moodle learning management system supports:

- activity completion tracking
- learning analytics
- automated notifications for incomplete tasks or approaching deadlines

These tools help both students and instructors monitor progress and intervene when necessary to support successful course completion.

Peer learning and collaboration

Although many assignments are completed individually, collaborative learning is also incorporated into the programme. The **Project Laboratory course in the first semester** requires students to work in teams on a joint development task.
In this course students may also participate in **peer-review activities**, which allow them to evaluate each other's work and receive feedback from fellow students. This approach supports collaborative learning and helps students develop communication and teamwork skills that are essential in professional engineering environments.

Role of campus weeks in student support

The on-campus intensive periods play an important role in strengthening student support. During these weeks students have the opportunity to meet instructors and supervisors in person, discuss their progress, and receive immediate feedback on their work.
These face-to-face interactions complement the online support system and help maintain a strong academic community within the programme.

Assessment and Quality Assurance in the Distance Learning Environment

The MSc programme in Computer Science Engineering applies a comprehensive assessment and quality assurance framework designed specifically for the distance learning environment. The framework ensures that student performance is evaluated reliably while maintaining academic integrity and supporting continuous improvement of the educational process.

Assessment strategy

Student performance in the programme is evaluated through a combination of continuous assessment, examinations, and project-based work. The assessment model is designed to measure both theoretical knowledge and practical engineering competences.
A typical assessment distribution in courses follows the structure below:

- **30% continuous assessment**, typically through quizzes and smaller assignments
- **30% examinations**, assessing theoretical understanding of the course material
- **40% project-based evaluation**, measuring the ability to apply knowledge in practical tasks

Continuous assessment encourages regular engagement with course materials and helps students receive feedback throughout the semester rather than only at its end.

Online assessment methods

In the asynchronous online learning environment, several digital assessment methods are used to evaluate student learning.
These include:

- automated quizzes within the Moodle learning management system
- programming assignments and computational tasks

- project-based assignments requiring independent implementation
- peer-review activities in selected courses

Automated quizzes provide immediate feedback to students and allow instructors to evaluate conceptual understanding efficiently.
Programming tasks and technical assignments allow students to demonstrate practical competence in applying engineering methods and technologies.

In-person examinations and supervised assessments

To ensure academic integrity and reliable evaluation of learning outcomes, certain assessment activities are conducted during the **on-campus examination week**.
These supervised activities may include:

- written examinations
- oral examinations
- project presentations and reports
- evaluations of laboratory or development tasks

The presence-based examination period ensures that key assessments are performed under controlled academic conditions.

Project-based evaluation

Project-based learning plays a central role in the programme, and therefore project evaluation constitutes a significant part of student assessment.
Projects are evaluated based on several criteria, including:

- correctness and functionality of the implemented solution
- quality of the design and implementation
- documentation and presentation of the results
- ability to apply appropriate technologies and engineering methods

In early project courses peer-review activities may complement instructor evaluation, allowing students to receive feedback from fellow participants. In advanced project courses and thesis work evaluation is conducted by instructors or supervisors.

Academic integrity

The programme applies several measures to maintain academic integrity in the distance learning environment.
These include:

- the use of supervised in-person examinations for critical assessments
- project-based assignments that require individual implementation
- question banks and randomized quizzes in the learning management system
- plagiarism detection tools where applicable

The combination of online and supervised in-person assessments helps ensure that evaluation results reliably reflect student knowledge and performance.

Monitoring learning progress

The Moodle learning management system provides several tools that support monitoring of student learning progress.
These include:

- activity completion tracking
- automated notifications for unfinished tasks
- learning analytics tools that identify students who may be falling behind

These tools allow instructors to detect potential learning difficulties early and offer additional support when necessary.

Continuous course evaluation

The quality of the programme is continuously monitored through several feedback mechanisms.
Students are encouraged to provide feedback on courses through institutional course evaluation surveys. This feedback is used by instructors and programme coordinators to improve course materials, teaching methods, and assessment strategies.
Instructors also review course performance indicators such as:

- completion rates
- assessment results
- student participation in learning activities

Based on these indicators, course materials and instructional methods may be updated to improve learning outcomes.

Continuous development of digital learning materials

The online learning materials used in the programme are regularly reviewed and updated to reflect advances in Computer Science Engineering and improvements in educational practices.

Course development activities include:

- updating lecture materials and examples
- improving clarity and structure of learning units
- increasing the use of interactive elements where appropriate
- incorporating feedback from previous cohorts of students

This continuous development process ensures that the programme maintains a high academic standard and remains aligned with current technological developments.

Digital Infrastructure of the Programme

The distance learning MSc programme in Computer Science Engineering is supported by an integrated digital learning environment that enables the delivery of course materials, communication between students and instructors, and the management of assessment activities. The infrastructure combines a learning management system with collaborative communication tools and discipline-specific software platforms required for Computer Science Engineering education.

Learning management system

The primary platform used for course delivery is the **Moodle learning management system** operated by the university. Moodle serves as the central environment for organizing course content and managing learning activities.

Within Moodle, students can access:

- lecture recordings and learning materials
- course descriptions and learning objectives
- quizzes and self-assessment tests
- programming and computational assignments
- project instructions and submission interfaces
- course announcements and discussion forums

Moodle also provides tools for **activity completion tracking and learning analytics**, which allow both students and instructors to monitor learning progress. These tools help identify incomplete tasks, approaching deadlines, and potential learning difficulties.

Communication and collaboration platform

Communication and online consultations are supported through **Microsoft Teams**, which is integrated into the university's digital infrastructure.
Teams is used for:

- course-related communication through dedicated channels
- online consultations and meetings between students and instructors
- project supervision and discussion
- online office hours when required

This platform enables both synchronous and asynchronous interaction and supports effective collaboration among students and instructors.

Support for programming and technical courses

As the programme focuses on Computer Science Engineering, many courses require the use of specialized software tools and programming environments. Students typically complete programming assignments and technical tasks on their own computers using widely available development tools and platforms.
Depending on the course, these may include:

- programming environments and development frameworks
- database management systems
- data analysis and visualisation tools
- simulation and modelling software

The use of commonly available tools ensures that students can complete course activities in a flexible learning environment while developing practical skills aligned with industry practices.

Digital submission and evaluation

Assignments and project deliverables are typically submitted through the Moodle platform. The system supports multiple forms of evaluation, including:

- automated quizzes and tests
- submission of programming tasks and project documentation
- instructor feedback and grading
- peer-review activities in selected courses

This digital workflow supports efficient course management and enables transparent feedback on student performance.

Accessibility and flexibility

The digital infrastructure is designed to support flexible participation in the programme. Learning materials can be accessed remotely at any time, allowing students to organise their learning schedule according to their individual circumstances.
The use of widely available software platforms and standard communication tools ensures that students can participate in the programme from different locations while maintaining full access to course resources and instructor support.

Reliability and institutional support

The digital systems used in the programme are operated and maintained by the university's IT services, ensuring reliable access to learning resources and communication tools.
Regular maintenance, system updates, and technical support help maintain the stability and security of the digital learning environment throughout the programme.

Student Workload and Learning Time Model

The MSc programme in Computer Science Engineering follows the principles of the **European Credit Transfer and Accumulation System (ECTS)**. In accordance with ECTS guidelines, one credit corresponds to approximately **30 hours of total student workload**, including both contact activities and independent learning.
The complete programme therefore represents an overall student workload of approximately **3600 hours**, distributed across four semesters and corresponding to a total of **120 ECTS credits**.

Components of student workload

Student workload in the distance learning environment consists of several complementary learning activities. These activities collectively support the acquisition of theoretical knowledge, practical engineering skills, and independent problem-solving abilities.

Typical workload components include:

- studying lecture recordings and learning materials
- reading supporting materials and documentation
- completing quizzes and self-assessment tasks
- solving programming and technical exercises
- working on individual and group projects
- participating in consultations with instructors or supervisors
- preparing for examinations
- completing project presentations and reports

These activities are distributed throughout the semester to support continuous learning and to prevent excessive workload accumulation toward the end of the study period.

Workload distribution within courses

Although the exact distribution may vary between courses depending on their learning objectives, the approximate workload distribution for a typical **6 ECTS course** (corresponding to about 180 hours of student work) may follow the pattern below:

Activity	Approximate workload
Studying lecture materials and videos	40–50 hours
Completing quizzes and smaller assignments	20–30 hours
Programming exercises and practical tasks	40–50 hours
Project work and implementation tasks	30–40 hours
Examination preparation	20–30 hours

This structure ensures that students engage regularly with the course material and develop both conceptual understanding and practical skills.

Workload distribution across the programme

The programme is designed so that students gradually transition from structured coursework to independent project-based work.

During the **first two semesters**, workload is primarily associated with coursework, programming exercises, and smaller project tasks supporting the acquisition of advanced theoretical and technical knowledge.

In the **third and fourth semesters**, a larger proportion of student workload is dedicated to the **capstone project and master's thesis**, which require independent problem solving, research, and development activities under academic supervision.

Self-paced learning and time management

Because the programme relies heavily on asynchronous learning, students can organise their learning schedule flexibly. However, courses provide recommended study schedules and assignment deadlines that help students distribute their workload evenly throughout the semester.

Learning management system features such as activity completion tracking and automated notifications also support effective time management and help students maintain steady progress.

Workload during the campus weeks

The on-campus intensive periods represent only a small portion of the overall workload of the programme. Activities during the campus weeks typically include:

- consultations with instructors and supervisors
- collaborative project work
- laboratory sessions where applicable
- in-person examinations or presentations

These activities complement the asynchronous learning process and provide opportunities for direct interaction and supervised assessment.

Alignment with learning outcomes

The workload model is designed to ensure that the time invested by students corresponds to the complexity of the learning outcomes defined for each course and for the programme.

The balance between theoretical study, practical assignments, and project-based work ensures that graduates of the programme acquire both advanced knowledge in Computer Science Engineering and the practical competences required for professional engineering practice and research activities.

Conclusion

The distance learning MSc programme in Computer Science Engineering at the University of Pannonia combines flexible online learning with structured academic supervision and periodic on-campus activities. The pedagogical model integrates asynchronous digital learning, project-based competence development, and supervised assessment in order to ensure that students achieve the intended learning outcomes of the programme.

The learning environment is built upon a robust digital infrastructure that supports the delivery of course materials, communication between students and instructors, and the management of assessment activities. Through the use of the Moodle learning management system and Microsoft Teams collaboration platform, students are able to access learning resources, complete assignments, and receive academic guidance regardless of their geographical location.

At the same time, the inclusion of intensive campus weeks during the study period and the examination period ensures that essential academic activities—such as consultations, project work, and key assessments—can take place in a supervised and interactive environment. This blended structure strengthens the reliability of assessment processes while also supporting direct academic interaction.

The programme gradually transitions students from structured coursework to increasingly independent project-based work. Through project laboratories, individual development tasks, and the master’s thesis, students develop the ability to apply advanced Computer Science Engineering knowledge to complex technological problems and applied research challenges.

Overall, the pedagogical framework of the programme provides a coherent and balanced learning environment that supports the development of both advanced theoretical knowledge and practical engineering competences required for professional practice or further academic research in Computer Science Engineering.