

Master of Science in Software Engineering

Program Objective

The Master's Program in Computer Software Engineering, administered by the School of Engineering, provides comprehensive coverage of both the theory and practice of software design and development in a project-based learning environment. It builds an understanding of the development lifecycle, including concept, design, testing, quality assurance (QA), and deployment. The graduate program in Computer Software Engineering advances the University's mission by supporting students' personal and professional growth so they can become effective contributors to the engineering profession and to society. The MS in Computer Software Engineering consists of 36 semester credit hours of core coursework.

Program Description

The program gives students the opportunity to develop their own big-picture understanding of the latest ideas, technologies, and methodologies. Collaborative work in a project-based setting enables students to share responsibilities across borders and cultures, and to reflect on real-world development processes. Before entering the program, students receive an introduction to the program and the University's online learning platform. The program is completed, on average, in 18 months.

Prospective Students

To be admitted to the Master's program in Software Engineering, applicants must meet certain entry requirements and demonstrate an appropriate background. Typical requirements include:

- **Bachelor's degree**: Applicants must hold a bachelor's degree in a related field, such as Software Engineering, Computer Science, Systems Engineering, or a similar discipline. The degree must be from a recognized, accredited institution.
- **Work experience**: While not always a strict requirement, prior professional experience in the field of software engineering may be viewed favorably during the admissions process. This experience may include work in software development, systems analysis, project management, or similar roles.
- **Technical knowledge:** Applicants are expected to have a strong foundation in concepts and technical skills related to software engineering. This may include knowledge of programming, software design, databases, operating systems, and software development methodologies.
- **Statement of purpose:** Applicants are typically required to submit a statement of purpose describing their motivations, goals, and expectations regarding the master's in software engineering. This provides a clearer picture of their interest in the field and their fit for the program.



Course Methodology

BIU University's study methodology focuses on two core pillars to maximize learning and real-world applicability.

- Neuroscience-based learning model: To achieve this, we leverage case studies to spark discussion, complete targeted tasks, and encourage students to support their viewpoints based on academic learning. In other words, BIU University doesn't rely on "lecture-style classes." Instead, we design live sessions (video conferencing and instructor follow-up) around presenting and debating case studies—much like what you'd see in a committee or board setting within companies or academic institutions.
- "Guided Online" academic coordination model: We recognize that both in-person and virtual formats come with strengths and drawbacks. That's why we've worked to reduce the highest-risk or highest-"impact" elements and strengthen everything in our methodology that is most effective and ensures learning in both modalities. Among the advantages of in-person learning are factors like stable groups, set schedules, and structured commitment; however, its main weakness is often how hard it can be to balance professional, personal, and academic life due to limited flexibility.

On the other hand, in the online model, flexibility is the standout advantage—but it also tends to come with a much higher dropout rate. That's why at BIU we don't believe in "pure online" models, where students are expected to independently consume the content they're given. For that reason, our academic model is explained through the following perspectives.

Regarding the course delivery platform:

The best of in-person learning environments

- A program with a planned calendar
- In fixed work groups
- Using collaborative teaching models
- With a bimodal workload (in class and outside of class)

The best of the online environment:

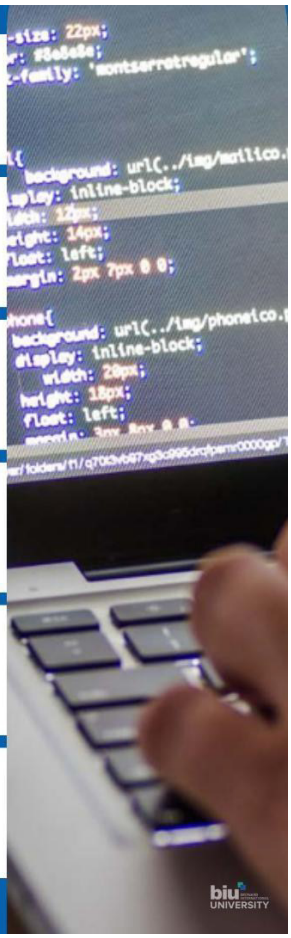
- Asynchronous learning and assignments that can be completed on a flexible schedule.
- Instant, virtual access to more than 250,000 sources of information through our virtual library.
- Easy access to the campus via Blackboard Ultra and Blackboard Collaborate.
-

Regarding the teaching method:

Bringing real workplace and/or company challenges into the classroom. With the support of the Student Care team and its proactive and reactive review of each student's academic progress at every stage—grounded in a rigorous KPI-tracking model—we deliver a high-performance, career-focused methodology with a dropout rate below 10%, well under the typical American university benchmark (around 20%).

	Nombre del curso	Código	Créditos
1° Período Académico	Advanced Computer Structures	CSE6011	3
	Aspects of Emerging computing	CSE6012	3
2° Período Académico	Innovative systems analysis and design	CSE6021	3
	Engineering modern software systems *	CSE6022	3
3° Período Académico	Strategic IT Project Management *	CSE6031	3
	Management of QA and Software Testing *	CSE6032	3
4° Período Académico	Research Methodology	CSE6051	3
	Next-Gen Web Development	CSE6042	3
5° Período Académico	Fundamentals of the new Object-Oriented Programming *	CSE6041	3
	Web Applications, Languages and Interchange Forms	CSE6052	3
6° Período Académico	Human-Computer Interaction, and Digital Citizenship	CSE6061	3
	Thesis	CSE6062	3

(*) Courses replacing the set of Concentration courses related to the standard MS in Computer Software Engineering.



CSE6011 - Advanced Computer Structures: The goal of this course is to provide a comprehensive overview of the core software and hardware technologies. It will give students a strong theoretical and hands-on foundation on which to build their understanding of future technical developments. The course will cover key concepts in computer structures, including hardware technologies and components, as well as the fundamentals of software systems and architectures.

Students will learn how software and hardware interact within a computer system and will build a deep understanding of how these essential components work and fit together. By the end of the course, students will be prepared to analyze and

understand the underlying structures and technologies in computer systems, enabling them to adapt and make sense of future technical advances more effectively.

CSE6012 - Aspects of Emerging computing: This course offers a broad understanding of the social and legal context in which information technology operates. Students will examine the relationship between information technology, society, and the law. The course helps build an understanding of external factors that affect computer systems and organizations, provides an overview of professional and ethical issues, and develops the skills needed to manage systems effectively while being mindful of their operating environment. Students will analyze ethical and legal challenges related to computing, understand the social implications of information technology, and gain the skills required to address and resolve professional issues in the computing field. By the end of the course, students will be prepared to make informed, ethical decisions in their professional computing practice.

CSE6021 - Innovative systems analysis and design: This course is designed to help students build the core skills needed to understand complex systems and challenges, and to develop effective automated solutions. The course takes a modern, object-oriented approach to modeling systems and producing software package designs that can automate those systems. It provides the skills needed to master these techniques, as well as the ability to use the Unified Modeling Language (UML) to describe these models.

Throughout the course, students will learn how to evaluate existing systems, define requirements, design effective solutions, and create UML-based models. The course also explores techniques and tools for user interface design and software lifecycle management. By the end of the course, students will be ready to apply systems analysis and design methods in software development settings and deliver automated solutions that are efficient and well designed.

CSE6022 - Engineering modern software systems: This course is designed to provide a strong theoretical foundation along with practical, hands-on software engineering skills. It covers both the theory and real-world practice of three core phases: problem definition, software development, and maintenance. Students will build the skills needed to lead a programming project and deliver products on time and within budget.

Throughout the course, students will explore a range of methodologies and techniques used in software engineering, including the software development life cycle, requirements management, system design and implementation of systems, software verification and validation of software, and software maintenance. Students will also learn best practices in project management and collaboration within development teams. By the end of the course, students will be ready to tackle software development challenges effectively and apply software engineering principles and techniques to achieve successful outcomes.

CSE6031 - Strategic IT Project Management: The purpose of this course is to provide a comprehensive understanding of the roles, responsibilities, and management techniques required for software technology projects. It demonstrates how technology project management supports the evolution of an information system from initial concept through implementation. Throughout the course, students will explore core project management fundamentals, including planning, monitoring and control, resource allocation, risk management, schedule management, and budgeting.

The course also covers IT project management topics such as scope management, quality management, communications management, and stakeholder management. Students will learn how to apply project management methodologies and frameworks—such as the PMBOK (Project Management Body of Knowledge) approach and Agile—within the context of software technology projects. By the end of the course, students will be prepared to effectively manage IT projects and address the specific challenges associated with software technology project management.

CSE6032 - Management of QA and Software Testing: The purpose of this course is to provide a thorough understanding of how to ensure software quality, including testing, maintenance, and effective management. This course provides the essential techniques to design and implement tests, conduct inspections, and use release and maintenance procedures. It also covers key aspects of managing the quality assurance process.

Throughout the course, students will learn about different approaches and methodologies for ensuring software quality, including white-box and black-box testing, unit testing, integration testing, system testing, and acceptance testing. Topics related to software quality management will also be explored, such as test planning and scheduling, defect and issue management, and the implementation of quality standards and best practices in software development. By the end of the course, students will be prepared to apply testing techniques and manage software quality effectively in a professional setting.

CSE6051 - Research Methodology: In this course, students will build the skills needed to design and structure a scientific research project that can serve as the foundation for a Master's thesis. To do so, they will learn and apply the core principles of the scientific method, including conceptual development, ethics, and the selection of qualitative, quantitative, or mixed approaches.

Students will identify and clearly define a research problem related to software engineering, develop research questions and objectives, conduct a background/literature review, and determine the methods and procedures needed to address the problem. They will also create a research proposal that includes the methodology, population or sample, data collection and analysis techniques, and the procedures used to evaluate results. The project must be completed in accordance with current APA guidelines and the academic standards established for scientific research.

CSE6042 - Next-Gen Web Development: The goal of this course is to equip students with the essential theoretical and hands-on tools to build advanced, content-rich websites. By the end of the course, students will be able to design and create an advanced website and will be ready to take on complex projects online. Throughout the course, students will learn the core foundations of web programming, including the programming languages and technologies used to develop websites. They will explore topics such as user interface design, content structure and organization, interactivity, security, and accessibility in web programming for websites. They will also gain practical skills in the use of specific tools and techniques for advanced web development, such as working with databases, integrating multimedia, and implementing interactive features.

Through projects and hands-on exercises, students will apply what they learn and develop advanced websites that meet industry standards and user needs. By the end of the course, students will be prepared to tackle challenges in the development of complex websites and will be able to apply their skills in the workplace or launch independent projects in the field of internet programming.

CSE6041 - Fundamentals of the new Object-Oriented Programming: This course offers both a theoretical foundation and hands-on practice in object-oriented programming and design. Students will build essential problem-solving and coding skills needed to write well-structured, object-oriented programs. They will also explore other key techniques—such as modern distributed systems and component technology—grounded in the ideas that have made object-oriented programming today's leading approach to software development.

Throughout the course, students will learn the core principles of object-oriented programming, including encapsulation, inheritance, and polymorphism. They will also become familiar with the main concepts and tools used in OOP, such as classes, objects, methods, attributes, and relationships between objects. Through practical exercises and projects, students will apply what they learn to solve problems and implement solutions using object-oriented programming. By the end of the course, students will be ready to design and develop robust, scalable programs using object-oriented principles.

CSE6052 - Web Applications, Languages and Interchange Forms: This course provides an overview of how XML is used and the role it plays in the next generation of e-commerce applications. This module builds an understanding of core XML technologies, the standardized development environment they support, and what that means for future internet application development. Throughout the course, students will explore the fundamentals of XML, including document syntax and the structure of XML documents, along with related technologies such as DTD (Document Type Definition) and XML Schema. The course also covers transforming and presenting XML data using XSL (Extensible Stylesheet Language). Students will learn to apply XML in web application development, including creating valid XML documents, manipulating and transforming XML data, and integrating XML with other web languages and technologies such as HTML, CSS, and JavaScript.

In addition, the course examines specific XML use cases in e-commerce applications, such as data exchange between systems, product catalog management, and implementing web services. Through hands-on projects, students will apply what they've learned in the design and development of web applications that use XML as a data format. By the end of the course, students will be ready to use XML effectively in the development of web applications and will understand the impact of XML on the future of internet applications.

CSE6061 - Human-Computer Interaction, and Digital Citizenship: The goal of this course is to build an understanding of how to design, evaluate, and develop user-friendly interactive application interfaces. It provides a strong foundation in human-computer interaction. The course is also designed to raise awareness of creative content and to promote a clearer understanding of the rights connected to it. Ultimately, it aims to instill in students an appreciation of the value that creative content holds for its creator, and to encourage personal respect for creative rights in ways that positively shape behaviors and perceptions around digitally delivered content. Throughout the course, students will explore core HCI concepts such as usability, user experience, user-centered design, and accessibility. They will learn techniques and methods for assessing and improving the usability of interactive application interfaces, as well as for designing interfaces that meet users' needs and expectations.

In addition, the course will address digital citizenship and intellectual property rights in the digital environment. Students will reflect on the ethical and legal issues involved in using and sharing creative content online, and they will be encouraged to develop a responsible, respectful approach to copyright and other aspects of digital citizenship. Through hands-on projects and discussion activities, students will apply their knowledge to the design and evaluation of interactive application interfaces, as well as to analysis and reflection on topics related to digital citizenship and intellectual property rights. By the end of the course, students will be equipped with the skills and knowledge needed

to design and develop user-friendly interactive app interfaces and understand the importance of ethics and copyright in the digital environment.

CSE6062 - Thesis: The thesis project serves as the capstone for the entire program. The student must successfully complete this requirement with a grade of B or higher in order to graduate. A strong project begins with a well-written proposal approved by the primary instructor. Once the proposal is approved, the student conducts research to support the work. This project should showcase the disciplinary competencies the student developed throughout the program. At the end of the project, the student presents their findings to the primary instructor. This final oral presentation is part of the project. APA (American Psychological Association) guidelines must be followed.