SOFTWARE ENGINEERING (BS)

With software applications of enormous size, complexity, and expense now prevalent in diverse domains, software engineering has never been as important a field as it is now. Students gain necessary talents to be successful in today's organizations, following current industry practices: designing and developing software; understanding and applying software development processes and methodologies in their work; leveraging software development tools used in the various phases of the development life cycle, and functioning as an effective member of a software development team or organization. Students develop their knowledge and skill through high-level electives where they write major projects in diverse areas such as client/server programming for the web, distributed programming for large clusters of processors, database programming, and markup language transformation. While working on modern applications with current software engineering practices such as Extreme Programming, students learn to analyze and apply good algorithms and other relevant tools.

Curriculum

Code	Title	Hours
Major Requirements		
Select one of the following:		3-4
MATH 131	Applied Calculus I	
MATH 161	Calculus I [1]	
Select one of the following:		3
ISSCM 241	Business Statistics	
PSYC 304	Statistics	
STAT 103	Fundamentals of Statistics	
STAT 203	Introduction to Probability & Statistics	
COMP 141	Introduction to Computing Tools and Techniques	3
COMP 163	Discrete Structures	3
or MATH 201	Introduction to Discrete Mathematics & Number Theory	
COMP 170	Introduction to Object-Oriented Programming	3
COMP 264	Introduction to Computer Systems	3
COMP 271	Data Structures I	3
COMP 272	Data Structures II	3
COMP 313	Object-Oriented Design	3
COMP 317	Social, Legal, and Ethical Issues in Computing	3
COMP 330	Software Engineering	3
Software Engineering Restricted Electives		
Select nine credit hours from the following:		9
COMP 332	Requirements Engineering	
COMP 335	Formal Methods in Software Engineering	
COMP 370	Software Quality	
COMP 371	Programming Languages	
COMP 373	Advanced Object-Oriented Programming	
COMP 382	Compiler Construction	
Practicum Capstone		
Select six credit hours from the following: [2]		6
COMP 312	Open Source Software Practicum	
COMP 390	Broadening Participation in STEM (Computing, Math & Science)	
COMP 391	Internship in Computer Science	
COMP 398	Independent Study	
Computer Science 300-Level Electives		
Select ten credit hours from any COMP 300-Level Courses [3]		10
Computer Science Free Elective		
Select one of the following:		3
COMP 125	Visual Information Processing	
COMP 150	Introduction to Computing	
COMP 300-Level Course [3]		
Total Hours		**61**

1. By arrangement with the Undergraduate Program Director, the extra credit from MATH 161 Calculus I may be applied towards the "Computer Science Free Electives" category.
2. See the details of registering in the links for each course. Students are encouraged to complete these credits during junior and senior years to draw on prior experience. Note:
   - COMP 312 is a 3-credit course
   - COMP 390 is limited to 3 total credits
   - COMP 391 and COMP 398 will usually be limited to 6 total credits each, but permission may sometimes be granted for more.
3. A list of available COMP 300-level courses can be found here (https://catalog.luc.edu/undergraduate/arts-sciences/computer-science/#coursestext).

Suggested Sequence of Courses

The below sequence of courses is meant to be used as a suggested path for completing coursework. An individual student's completion of requirements depends on course offerings in a given term as well as the start term for a major or graduate study. Students should consult their advisor for assistance with course selection.

Course	Title	Hours
Year 1		
Fall		
COMP 141	Introduction to Computing Tools and Techniques	3
COMP 150	Introduction to Computing [1]	3
MATH 131	Applied Calculus I [2]	3
UNIV 101	First Year Seminar	1
CORE: College Writing Seminar		3
CORE: Philosophical Knowledge Tier 1		3
	Hours	**16**
Spring		
COMP 163	Discrete Structures	3
COMP 170	Introduction to Object-Oriented Programming [3]	3
STAT 103	Fundamentals of Statistics [4]	3
CORE: Ethics		3
CORE: Historical Knowledge Tier 1		3
	Hours	**15**

Year 2		
Fall		
COMP 264	Introduction to Computer Systems	3
COMP 271	Data Structures I	3
COMP 317	Social, Legal, and Ethical Issues in Computing	3
CAS Language Requirement 101 level [5]		3
CORE: Theology and Religious Studies Tier 1		3
	Hours	**15**
Spring		
COMP 272	Data Structures II	3
COMP 330	Software Engineering	3
CAS Language Requirement 102 level		3
CORE: Scientific Knowledge Tier 1		3
CORE: Societal & Cultural Knowledge Tier 1		3
	Hours	**15**
Year 3		
Fall		
COMP Free Elective		1
COMP Free Elective		3
CORE: Artistic Knowledge & Experience		3
CORE: Literary Knowledge & Experience Tier 1		3
CORE: Philosophical Knowledge Tier 2		3
Software Engineering Restricted Elective		3
	Hours	**16**
Spring		
COMP 313	Object-Oriented Design	3
CORE: Historical Knowledge Tier 2		3
CORE: Scientific Knowledge Tier 2		3
CORE: Theology and Religious Studies Tier 2		3
Software Engineering Restricted Elective		3
	Hours	**15**
Year 4		
Fall		
CAS Elective		3
COMP Practicum		3
CORE: Literary Knowledge & Experience Tier 2		3
CORE: Societal & Cultural Knowledge Tier 2		3
Software Engineering Restricted Elective		3
	Hours	**15**
Spring		
CAS Elective		3
COMP Free Elective		3
COMP Free Elective		3
COMP Free Elective if COMP 150 not taken		3
COMP Practicum		3
	Hours	**15**
	Total Hours	**122**

1 COMP 150 Introduction to Computing will apply to COMP Free Electives; students with prior experience in computer programming, for example a high school course modeled on the Exploring Computer Science (https://www.exploringcs.org/) or Computer Science Principles (https://apcentral.collegeboard.org/courses/ap-computer-science-principles/) curriculum may replace this course with a different COMP Free Elective at any time during the program. A score of 4 or 5 on the AP CS Principles Exam will earn actual credit for this course.

2 May substitute MATH 161 Calculus I and may use the extra credit towards COMP Free Electives.

3 A score of 4 or 5 on the AP CS A Exam will earn credit for this course.

4 May substitute STAT 203 Introduction to Probability & Statistics or ISSCM 241 Business Statistics or PSYC 304 Statistics.

5 Language must be completed through the 102 course level or through an exam (https://www.luc.edu/cas/college-requirements/).

General Notes

- Credits never can be double-counted for different categories of the requirements for the major. But a course may satisfy a major requirement and also satisfy a University and/or College requirement (e.g., Core, residency, Engaged Learning, Writing Intensive).
- With permission, extra credits of MATH 161 Calculus I or 300 level MATH, PHYS, or STAT **for double majors** can be applied to the "Computer Science 300-Level Electives" or "Computer Science Free Electives" categories.)
- It is usually not meaningful to combine a computing major or minor with another, the principal exception being Computer Crime and Forensics Minor (https://catalog.luc.edu/undergraduate/arts-sciences/computer-science/computer-crime-forensics-minor/); see more detail in the double-dipping rules (https://catalog.luc.edu/undergraduate/arts-sciences/computer-science/#policiestext).

College of Arts and Sciences Graduation Requirements

All Undergraduate students in the College of Arts and Sciences are required to take two Writing Intensive courses (6 credit hours) as well as complete a foreign language requirement at 102-level or higher (3 credit hours) or a language competency test. More information can be found here (https://www.luc.edu/cas/college-requirements/).

Additional Undergraduate Graduation Requirements

All Undergraduate students are required to complete the University Core, at least one Engaged Learning course, and UNIV 101. SCPS students are not required to take UNIV 101. Nursing students in the Accelerated BSN program are not required to take core or UNIV 101. You can find more information in the University Requirements (https://catalog.luc.edu/undergraduate/university-requirements/) area.

Learning Outcomes

1. Knowledge of Software Development Lifecycle: Students should understand the various stages of software development, from requirements elicitation, to design, implementation, testing, and maintenance.
2. Proficiency in Programming: Graduates should be proficient in several programming languages and have a deep understanding of object-oriented design and other software paradigms.
3. Software Design Skills: Graduates should be able to design, implement, validate, and maintain software systems. This includes the ability to work with complex software architectures and design patterns.

4. Understanding of Software Quality Assurance: This includes knowledge of testing methodologies, debugging, and techniques to ensure software reliability, usability, security, and performance.
5. Project Management Skills: Students should understand software project planning and management techniques. This includes knowledge of cost estimation, risk management, project scheduling and tracking.
6. Teamwork and Communication: Similar to computer science, students should be able to work effectively on teams and be able to communicate their ideas and work effectively both verbally and in writing.
7. Ethical and Professional Responsibility: Graduates should understand professional, ethical, legal, and societal responsibilities related to software engineering.