Narrative for the Revised B.S. in Computer Science Curriculum

Knight Foundation School of Computing and Information Sciences – Florida International University

1. Motivation and Strategic Context

The proposed revision to the B.S. in Computer Science curriculum is a direct response to three converging forces:

- The rapid evolution of computing practice, particularly the widespread adoption of AI-assisted tools in software development;
- Shifts in industry expectations, where graduates are expected to demonstrate not only programming proficiency, but also strong computational thinking, system-level understanding, and the ability to critically evaluate AI-generated artifacts; and
- Institutional priorities at FIU, including the deliberate integration of AI across the curriculum while maintaining rigor, ethical responsibility, and alignment with ABET accreditation standards.

The revised curriculum modernizes the degree while preserving the theoretical and systems foundations that define computer science as a discipline. Rather than introducing ad hoc changes, the redesign follows a cohesive, scaffolded approach that strengthens the core curriculum and provides students with more intentional flexibility in their elective pathways.

2. Strengthening the Core: 45 Credits of Required Courses

The revised program consists of **45 required credits**, ensuring that every graduate completes a coherent and comprehensive computer science core before specializing through electives.

a. Addition of Database Management (COP 4710)

Database systems are no longer optional knowledge for modern computer scientists. They underpin nearly every contemporary application domain, including web systems, data science, cybersecurity, enterprise software, and AI-driven applications.

By adding **Database Management (COP 4710)** as a required course, the curriculum ensures that all students:

- Understand data modeling, normalization, and query optimization;
- Can critically evaluate and validate SQL and database logic, including AI-generated queries; and
- Are better prepared for upper-level coursework in systems, software engineering, and web development.

This change closes a long-standing gap in the core curriculum and aligns FIU's CS degree with peer institutions nationally.

b. Introduction of Web Systems Development as a Required Course

A new required course, **Web Systems Development (COP 4849)**, has been designed to reflect how modern software systems are actually built, deployed, and maintained.

Unlike traditional "web programming" courses, this course emphasizes:

- Full-stack system design (front-end, back-end, databases, and APIs);
- Secure client–server communication and applied cryptography;
- Cloud deployment, CI/CD pipelines, and collaborative development workflows; and
- Responsible use of generative AI to accelerate—rather than replace—professional development practices.

Making this course required ensures that every CS graduate leaves FIU with hands-on experience designing and deploying a complete software system, a competency increasingly expected by employers and graduate programs alike.

3. A Deliberate Programming Language Progression

The revised curriculum adopts a purpose-driven, language-conscious progression, rather than tying learning outcomes too tightly to any single programming language.

- **COP 2047 (Programming I)** uses **Python**
  Python lowers the syntactic barrier for beginners, allowing students to focus on problem-solving, decomposition, and algorithmic thinking—skills that are even more critical in an AI-assisted programming landscape.
- **COP 3337 (Programming II)** remains in **Java**
  Java provides a strong foundation in object-oriented design, abstraction, and type discipline, preparing students for large-scale systems and industry-standard software development practices.
- **COP 4338 (Systems Programming)** remains in **C**
  Retaining C ensures students gain low-level systems knowledge, explicit memory management expertise, and exposure to Unix-based development environments.
- **Other core courses** (e.g., Data Structures, Programming Languages, Software Engineering) remain **language-agnostic**, reinforcing transferable concepts rather than tool-specific skills and aligning with ABET expectations.

This progression balances accessibility, rigor, and depth while ensuring graduates are fluent across multiple programming paradigms.

4. Evidence-Based Rationale for Moving COP 2047 (CS1) from Java to Python

A. Market Signals from Widely Cited Programming Language Indices

Multiple widely cited programming language indices consistently rank **Python as the most popular programming language worldwide**, with a significant margin over Java.

Table 1. Programming Language Popularity Snapshot in September 2025 (see www.tiobe.com/tiobe-index)

Rank	Language	Rating	Year-over-Year Change
1	Python	25.98%	+5.81%
2	C++	8.80%	−1.94%
3	C	8.65%	−0.24%
4	Java	8.35%	−1.09%
5	C#	6.38%	+0.30%
6	JavaScript	3.22%	−0.70%
7	Visual Basic	2.84%	+0.14%

These signals matter for curriculum design because they:

- Support student motivation and perceived relevance early in the major; and
- Align the CS1 experience with the languages students encounter in industry, AI, data science, automation, scripting, and modern web back-end development.

B. Long-Term Trends in Language Adoption

Longitudinal data over the past two decades shows sustained growth in Python's adoption and usage, particularly in recent years. While Java remains an important and widely used language, its relative popularity has plateaued compared to Python's trajectory.

This trend supports a curriculum strategy in which:

- **CS1 leverages Python** for early momentum, conceptual clarity, and broad applicability; and
- **CS2 retains Java** to deepen object-oriented design, static typing discipline, and software engineering practices.

This sequencing reflects a deliberate pedagogical distinction: **CS1 emphasizes foundational problem-solving, while CS2 emphasizes software engineering rigor.**

C. Pedagogical Fit: Improving the Signal-to-Noise Ratio in CS1

Python is particularly well suited for an introductory programming course that emphasizes computational thinking. Compared to Java, Python offers:

- Simpler syntax and higher readability;
- A gentler learning curve for novice programmers;
- Immediate feedback and reduced early frustration;
- No exposure to manual memory management in CS1; and
- Strong relevance across modern application domains, including AI and data science.

By reducing incidental complexity, Python allows COP 2047 to better assess and develop the outcomes that matter most early in the curriculum:

- Problem decomposition;
- Algorithmic reasoning;
- Correctness, testing, and debugging habits; and
- Ethical and responsible computing practices, including AI-assisted workflows.

This change also complements the addition of **COP 3410 (Computational Thinking)**, which explicitly strengthens problem-solving skills early in the program.

D. Peer Benchmarking Across Public Universities

A growing number of large public universities have transitioned their introductory computer science sequence from Java or C++ to Python, often while retaining Java or C++ in later courses.

Table 2. Examples of CS1 Transitions to Python at Peer Institutions

University	Prior CS1 Language	Transition Period	Current CS1 Model
University of Florida	Java	2022	Python-based CS1
Penn State	C++	2021	Python-based intro sequence
UC San Diego	Java	2018 (phased)	Python-first introductory track

UC Irvine	Java (C++ prior)	2017–2018	Python-based CS1
University of Virginia	Java	2019 (phased)	Python-first intro pathway
UMass Amherst	Java	Announced 2022	Python-based CS1

Importantly, many of these institutions retain Java, C, or C++ in second-year and systems courses, reinforcing a progression similar to FIU's proposed structure.

E. Why Language Choice Matters More in the Generative AI Era

A common concern is whether programming language choice still matters when students can use generative AI tools to produce code. In practice, generative AI increases the importance of early conceptual mastery rather than diminishing it.

Python in CS1 makes it easier for students to:

- Practice decomposition and algorithmic thinking without syntax overhead;
- Iteratively test and debug solutions;
- Meaningfully compare human-written and AI-generated solutions; and
- Develop responsible-use habits, including validation, transparency, and attribution.

At the same time, retaining Java in CS2 preserves critical software engineering discipline, including:

- Encapsulation and interface-driven design;
- Static typing and reasoning about larger codebases; and
- Industry-standard development practices.

Together, the sequence reinforces a clear distinction:
CS1 focuses on computational thinking; CS2 focuses on software engineering.

5. Elevating Computational Thinking: COP 3410

A key addition to the core curriculum is **Computational Thinking (COP 3410)**, introduced as a required course and a co-requisite for Programming II.

The motivation for this course is twofold:

- AI-assisted tools have changed what it means to "know how to program"; students must be able to reason about, evaluate, and refine solutions—not merely produce code.
- Problem-solving, abstraction, and decomposition are now first-order skills that must be explicitly taught rather than implicitly assumed.

COP 3410 strengthens student readiness for Data Structures, Algorithms, Systems, and AI-integrated coursework by emphasizing:

- Algorithmic reasoning independent of syntax;
- Problem decomposition and representation; and
- Ethical and responsible use of AI as a computational tool.

6. AI Integration: Intentional, Ethical, and Assessable

Rather than isolating AI in a single elective, AI-assisted tools have been integrated across the majority of required courses in a principled and assessable manner.

Across the core, students are expected to:

- Use AI tools for code explanation, debugging, test generation, and design ideation;
- Critically evaluate AI-generated outputs for correctness, efficiency, security, and bias; and
- Reflect on ethical responsibilities, limitations, and appropriate reliance on AI.

AI is treated consistently as **a tool to be analyzed, validated, and leveraged—not a shortcut that replaces learning**. This approach aligns with FIU's broader AI strategy while preserving academic integrity and rigor.

7. Reimagined Electives: Flexibility Without Sacrificing Breadth

The elective structure has been reorganized into four thematic categories:

- Foundations (at-least 3 credits)
- Systems (at-least 3 credits)
- Applications (at-least 3 credits)
- CIS4947 (Career-Readiness) or CIS 4912 (REU) or CIS 3590 (at-most 6 credits)
- Total: 15 credits

8. Program-Wide Alignment Across CS Degree Pathways

While the primary focus of this proposal is the revision of the **B.S. in Computer Science (main track)**, the curricular changes have been **systematically propagated across all undergraduate CS pathways** to ensure consistency, coherence, and alignment.

Specifically, the revised core structure, sequencing, and pedagogical principles reflected in the B.S. in CS main track are also incorporated into:

- the **B.S. in Computer Science – Software Design and Development (SDD) Track**,
- the **Bachelor of Arts in Computer Science**, and
- the **Minor in Computer Science**.

Across these programs, shared foundational elements, including the revised programming sequence, the addition of **Computational Thinking (COP 3410)**, the inclusion of **Database Management (COP 4710)** where appropriate, and the integration of **responsible generative AI use**—are consistently reflected in the corresponding flowcharts.

At the same time, each program retains its **distinct identity and degree objectives**:

- the SDD track emphasizes applied software development depth,
- the B.A. in CS preserves flexibility and interdisciplinary breadth, and
- the CS minor provides a coherent and modern introduction to computing for non-majors.

This coordinated update ensures that students across all CS-related programs benefit from a **shared, modernized foundation**, while preserving the intentional differences in

scope, depth, and outcomes appropriate to each degree. The attached one-page flowcharts for each program visually illustrate this alignment and confirm that the proposed changes are **program-wide, intentional, and internally consistent**.

9. Summary and Value Proposition

Taken together, the revised B.S. in Computer Science curriculum:

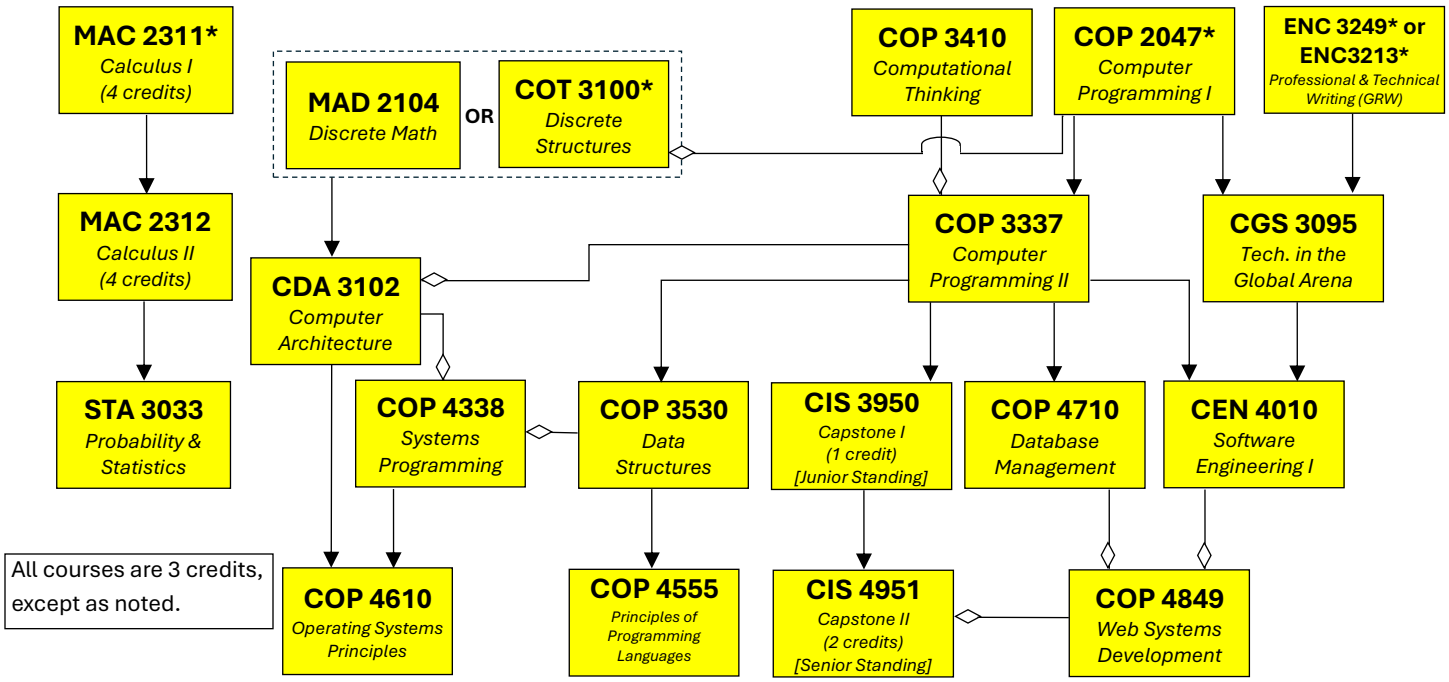
- Modernizes the degree without diluting foundational rigor;
- Aligns with industry practice, graduate expectations, and ABET standards;
- Integrates AI-assisted tools responsibly across the curriculum;
- Strengthens problem-solving and computational thinking; and
- Provides students with both breadth and flexibility.

Most importantly, it positions FIU CS graduates not merely as users of current tools, but as adaptable computer scientists prepared for the next decade of technological change.

FIU | Knight Foundation School of Computing and Information Sciences

Bachelor of Science in Computer Science

Computer Science Track



CS Electives (15 credits): **Must take at-least one course from each group**. Remaining credits must be taken *either from these three elective groups OR at-most 6 credits from the group (CIS 4947, CIS 4912, CIS 3590)*:

Foundations

- CAP 4506 – Intro to Game Theory (Prereq: MAC2312)
- COP 4534 – Algorithm Techniques (Prereq: COP3530)
- COT 3510 – Applied Linear Structures (Prereq: see syllabus; Coreq: COT3100 or MAD2104)
- COT 3541 – Logic for CS (Prereq: COP3337 & COT3100)
- COT 4521 – Intro to Computational Geometry (Prereq: COP3530)
- COT 4601 – Fundamentals of Quantum Computing (Prereq: (COP3337 or COP3804) and COT3100)
- MAD 3301 – Graph Theory (Prereq: COP2210 & (COT3100 or MAD2104))
- MAD 3401 – Numerical Analysis (Prereq: COP2210, MAC2312)
- MAD 3512 – Theory of Algorithms (Prereq: COP3530)
- MAD 4203 – Combinatorics (Prereq: MAD2104, MAC2312)
- MHF 4302 – Math Logic (Prereq: MAD3512)

Systems

- CAP 4453 – Robot Vision (Prereq: COP3530, MAC2312)
- CDA 4625 – Intro to Mobile Robotics (Prereq: COP3530, STA3033)
- CNT 4713 – Net-Cent. Computing (Prereq: COP4338)
- CIS 4203 – Digital Forensics (Prereq: COP2210/COP2250/EEL2880)
- CIS 4731 – Fund Blockchain Tech. (Prereq: (COP2047 OR COP2210) AND COT3100)
- COP 4520 – Intro to Parallel Computing (Prereq: COP3530 and (CDA3102 or CDA4101 or EEL4709))
- COP 4604 – Advanced UNIX Programming (Prereq: COP4610)
- COP 4751 – Advanced Database Management (Prereq: COP4710)
- CTS 4408 – Database Administration (Prereq: COP4710 or COP4703)
- COT 4431 – Applied Parallel Computing (Prereq: COP3530 and (CDA3102 or EEL4709))

Applications

- CAI 4002 – Artificial Intelligence (Prereq: COP3530)
- CAI 4105 – Introduction to Machine Learning (Prereq: COP3530, STA3033)
- CAI 4304 – Nat Lang Processing (Prereq: COP3530)
- CAP 4052 – Game Design & Dev (Prereq: (STA2023 or STA3033) and COP3530 and CAP4104)
- CAP 4104 – Human Cmptr Interaction (Prereq: COP3337)
- CAP 4710 – Computer Graphics (Prereq: COP3337, MAC2312)
- CAP 4770 – Intro to Data Mining (Prereq: STA-2023/2122/3033/4322 and COP-3530/3465)
- CAP 4830 – Modeling & Simulations (Prereq: (STA2023 or STA3033) and COP3530)
- CEN 4021 – Software Engineering II (Prereq: CEN4010)
- CEN 4072 – Software Testing (Prereq: COP3530)
- COP 4226 – Adv Windows Prog. (Prereq: COP3530)
- COP 4655 – Mobile App Dev (Prereq: CAP4104 and CEN4010)

* Pre-requisites:

- COP2047: MAC1140 or higher level math course
- COT3100: Prereq: any MAC course and any COP course, Coreq:(COP2210 or COP2250 or EEL2280)
- ENC3249/ENC3213: UCC English requirements
- MAC2311: MAC1147
- MAD2104: MAC1105 or MGF1106

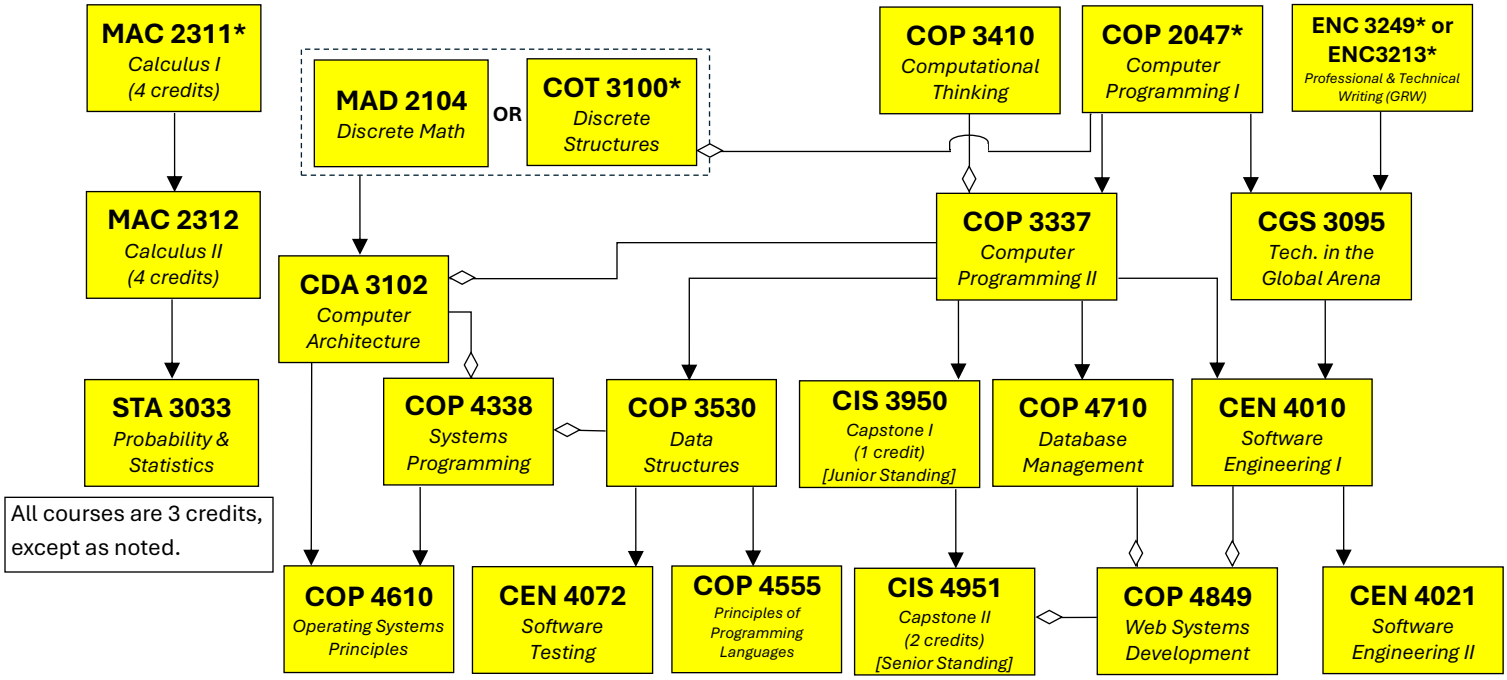
A direction line indicates a prerequisite. The course above must be completed before the course below can be taken.

A diamond indicates a co-requisite. The course closer to the diamond may be taken at the same time as the co-requisite. The co requisite is a prerequisite for any course that requires the course closer to the diamond.

FIU | Knight Foundation School of Computing and Information Sciences

Bachelor of Science in Computer Science

Software Design and Development Track



CS Electives (9 credits): **Must take at-least one course Foundations group and once course from Systems group**.
Remaining credits must *either be taken from these groups OR from the group (CIS 4947, CIS 4912, CIS 3590)*:

Foundations

- CAP 4506 – Intro to Game Theory (Prereq: MAC2312)
- COP 4534 – Algorithm Techniques (Prereq: COP3530)
- COT 3510 – Applied Linear Structures (Prereq: see syllabus; Coreq: COT3100 or MAD2104)
- COT 3541 – Logic for CS (Prereq: COP3337 & COT3100)
- COT 4521 – Intro to Computational Geometry (Prereq: COP3530)
- COT 4601 – Fundamentals of Quantum Computing (Prereq: (COP3337 or COP3804) and COT3100)
- MAD 3301 – Graph Theory (Prereq: COP2210 & (COT3100 or MAD2104))
- MAD 3401 – Numerical Analysis (Prereq: COP2210, MAC2312)
- MAD 3512 – Theory of Algorithms (Prereq: COP3530)
- MAD 4203 – Combinatorics (Prereq: MAD2104, MAC2312)
- MHF 4302 – Math Logic (Prereq: MAD3512)

Systems

- CAP 4453 – Robot Vision (Prereq: COP3530, MAC2312)
- CDA 4625 – Intro to Mobile Robotics (Prereq: COP3530, STA3033)
- CNT 4713 – Net-Cent. Computing (Prereq: COP4338)
- CIS 4203 – Digital Forensics (Prereq: COP2210/COP2250/EEL2880)
- CIS 4731 – Fund Blockchain Tech. (Prereq: (COP2047 OR COP2210) AND COT3100)
- COP 4520 – Intro to Parallel Computing (Prereq: COP3530 and (CDA3102 or CDA4101 or EEL4709))
- COP 4604 – Advanced UNIX Programming (Prereq: COP4610)
- COP 4751 – Advanced Database Management (Prereq: COP4710)
- CTS 4408 – Database Administration (Prereq: COP4710 or COP4703)
- COT 4431 – Applied Parallel Computing (Prereq: COP3530 and (CDA3102 or EEL4709))

Applications

- CAI 4002 – Artificial Intelligence (Prereq: COP3530)
- CAI 4105 – Introduction to Machine Learning (Prereq: COP3530, STA3033)
- CAI 4304 – Nat Lang Processing (Prereq: COP3530)
- CAP 4052 – Game Design & Dev (Prereq: (STA2023 or STA3033) and COP3530 and CAP4104)
- CAP 4104 – Human Cmptr Interaction (Prereq: COP3337)
- CAP 4710 – Computer Graphics (Prereq: COP3337, MAC2312)
- CAP 4770 – Intro to Data Mining (Prereq: STA-2023/2122/3033/4322 and COP-3530/3465)
- CAP 4830 – Modeling & Simulations (Prereq: (STA2023 or STA3033) and COP3530)
- CEN 4021 – Software Engineering II (Prereq: CEN4010)
- CEN 4072 – Software Testing (Prereq: COP3530)
- COP 4226 – Adv Windows Prog. (Prereq: COP3530)
- COP 4655 – Mobile App Dev (Prereq: CAP4104 and CEN4010)

* Pre-requisites:

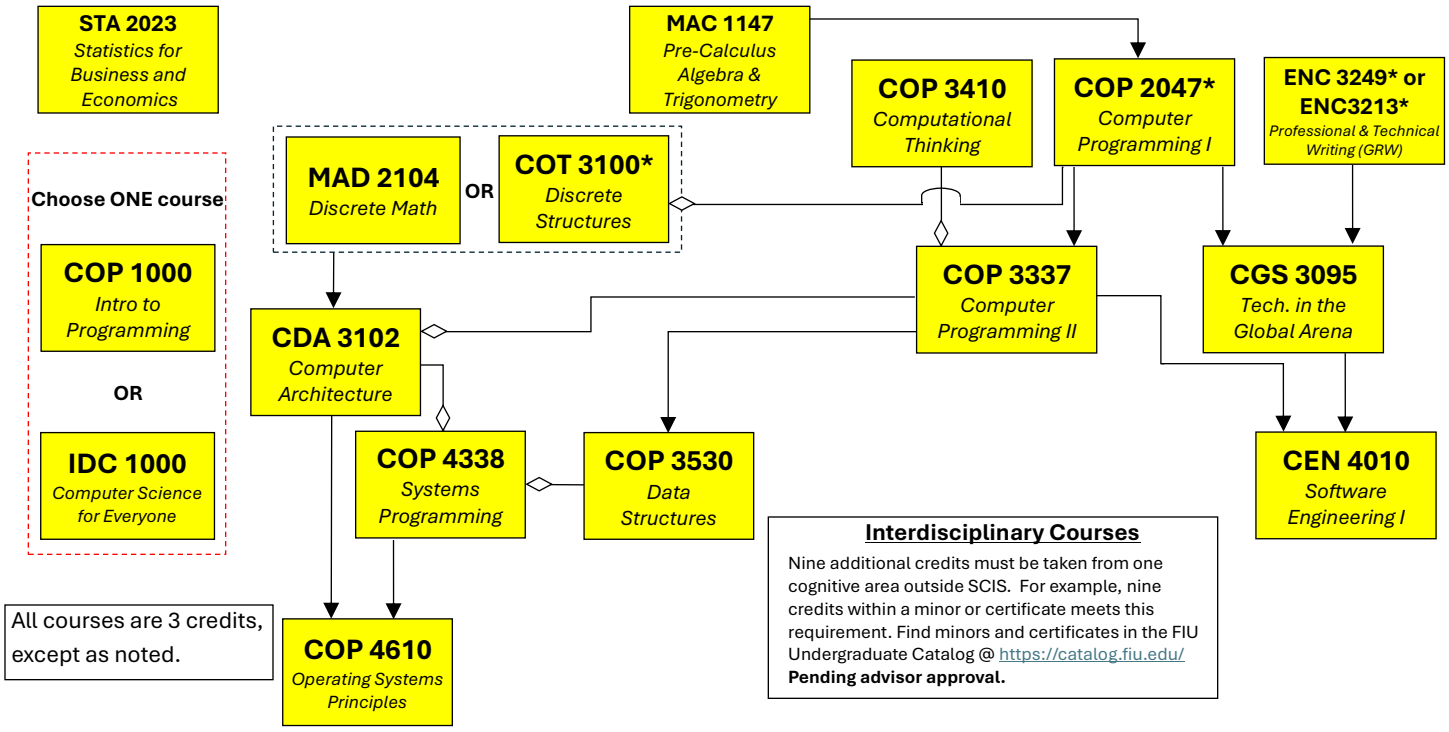
- COP2047: MAC1140 or higher-level math course
- COT3100: Prereq: any MAC course and any COP course, Coreq:(COP2210 or COP2250 or EEL2280)
- ENC3249/ENC3213: UCC English requirements
- MAC2311: MAC1147
- MAD2104: MAC1105 or MGF1106

A direction line indicates a prerequisite. The course above must be completed before the course below can be taken.

A diamond indicates a co-requisite. The course closer to the diamond may be taken at the same time as the co-requisite. The co requisite is a prerequisite for any course that requires the course closer to the diamond.

FIU Knight Foundation School of Computing and Information Sciences

Bachelor of Arts in Computer Science



All courses are 3 credits, except as noted.

Interdisciplinary Courses

Nine additional credits must be taken from one cognitive area outside SCIS. For example, nine credits within a minor or certificate meets this requirement. Find minors and certificates in the FIU Undergraduate Catalog @ https://catalog.fiu.edu/
Pending advisor approval.

CS Electives (15 credits): **Must take at-least one course from each group**. Remaining credits must be taken *either from these three elective groups OR at-most 6 credits from the group (CIS 4947, CIS 4912, CIS 3590)*:

Foundations

- CAP 4506 – Intro to Game Theory (Prereq: MAC2312)
- COP 4534 – Algorithm Techniques (Prereq: COP3530)
- COT 3510 – Applied Linear Structures (Prereq: see syllabus; Coreq: COT3100 or MAD2104)
- COT 3541 – Logic for CS (Prereq: COP3337 & COT3100)
- COT 4521 – Intro to Computational Geometry (Prereq: COP3530)
- COT 4601 – Fundamentals of Quantum Computing (Prereq: (COP3337 or COP3804) and COT3100)
- MAD 3301 – Graph Theory (Prereq: COP2210 & (COT3100 or MAD2104))
- MAD 3401 – Numerical Analysis (Prereq: COP2210, MAC2312)
- MAD 3512 – Theory of Algorithms (Prereq: COP3530)
- MAD 4203 – Combinatorics (Prereq: MAD2104, MAC2312)
- MHF 4302 – Math Logic (Prereq: MAD3512)

Systems

- CDA 4625 – Intro to Mobile Robotics (Prereq: COP3530, STA3033)
- CNT 4713 – Net-Cent. Computing (Prereq: COP4338)
- CIS 4203 – Digital Forensics (Prereq: COP2210/COP2250/EEL2880)
- CIS 4731 – Fund Blockchain Tech. (Prereq: (COP2047 OR COP2210) AND COT3100)
- COP 4520 – Intro to Parallel Computing (Prereq: COP3530 and (CDA3102 or CDA4101 or EEL4709))
- COP 4604 – Advanced UNIX Programming (Prereq: COP4610)
- COP 4751 – Advanced Database Management (Prereq: COP4710)
- CTS 4408 – Database Administration (Prereq: COP4710 or COP4703)
- COT 4431 – Applied Parallel Computing (Prereq: COP3530 and (CDA3102 or EEL4709))

Applications

- CAI 4002 – Artificial Intelligence (Prereq: COP3530)
- CAI 4105 – Introduction to Machine Learning (Prereq: COP3530, STA3033)
- CAI 4304 – Nat Lang Processing (Prereq: COP3530)
- CAP 4052 – Game Design & Dev (Prereq: (STA2023 or STA3033) and COP3530 and CAP4104)
- CAP 4104 – Human Cmptr Interaction (Prereq: COP3337)
- CAP 4710 – Computer Graphics (Prereq: COP3337, MAC2312)
- CAP 4770 – Intro to Data Mining (Prereq: STA-2023/2122/3033/4322 and COP-3530/3465)
- CAP 4830 – Modeling & Simulations (Prereq: (STA2023 or STA3033) and COP3530)
- CEN 4021 – Software Engineering II (Prereq: CEN4010)
- CEN 4072 – Software Testing (Prereq: COP3530)
- COP 4226 – Adv Windows Prog. (Prereq: COP3530)
- COP 4655 – Mobile App Dev (Prereq: CAP4104 and CEN4010)

* Pre-requisites:

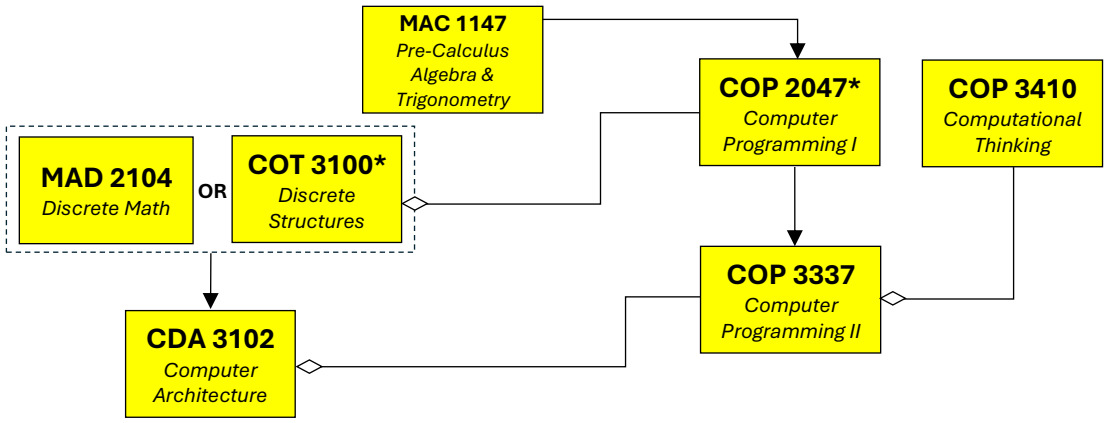
- COP2047: MAC1140 or higher level math course
- COT3100: Prereq: any MAC course and any COP course, Coreq:(COP2210 or COP2250 or EEL2280)
- ENC3249/ENC3213: UCC English requirements
- MAC2311: MAC1147
- MAD2104: MAC1105 or MGF1106

A direction line indicates a prerequisite. The course above must be completed before the course below can be taken.

A diamond indicates a co-requisite. The course closer to the diamond may be taken at the same time as the co-requisite. The co requisite is a prerequisite for any course that requires the course closer to the diamond.

FIU Knight Foundation School of Computing and Information Sciences

Minor in Computer Science



All courses are 3 credits, except as noted.

Interdisciplinary Courses

Nine additional credits must be taken from one cognitive area outside SCIS. For example, nine credits within a minor or certificate meets this requirement. Find minors and certificates in the FIU Undergraduate Catalog @ https://catalog.fiu.edu/
Pending advisor approval.

CS Minor Electives (6 credits): Choose from all courses below:

Foundations

- COP 4534 – Algorithm Techniques (Prereq: COP3530)
- COP 4555 – Principles of Programming Languages (Prereq: COP3530)
- COT 3510 – Applied Linear Structures (Prereq: see syllabus; Coreq: COT3100 or MAD2104)
- COT 3541 – Logic for CS (Prereq: COP3337 & COT3100)
- COT 4521 – Intro to Computational Geometry (Prereq: COP3530)
- COT 4601 – Fundamentals of Quantum Computing (Prereq: (COP3337 or COP3804) and COT3100)

Systems

- CIS 4203 – Digital Forensics (Prereq: COP2210/COP2250/EEL2880)
- CIS 4731 – Fund Blockchain Tech. (Prereq: (COP2047 OR COP2210) AND COT3100)
- COP 3530 – Data Structures (Prereq: COP3337 and (MAD2104 or COT3100))
- COP 4338 – Systems Programming (Coreq: CDA3102 and COP3530)
- COP 4520 – Intro to Parallel Computing (Prereq: COP3530 and (CDA3102 or CDA4101 or EEL4709))
- COP 4710 – Database Management (Prereq: COP3337)
- COT 4431 – Applied Parallel Computing (Prereq: COP3530 and (CDA3102 or EEL4709))

Applications

- CAI 4002 – Artificial Intelligence (Prereq: COP3530)
- CAP 4104 – Human Cmptr Interaction (Prereq: COP3337)
- CAP 4641 – Nat Language Processing (Prereq: COP3530)
- CAP 4830 – Modeling & Simulations (Prereq: (STA2023 or STA3033) and COP3530)
- CEN 4010 – Software Engineering I (Prereq: CGS3095 and COP3337)
- CEN 4021 – Software Engineering II (Prereq: CEN4010)
- CEN 4072 – Software Testing (Prereq: COP3530)
- COP 4226 – Adv Windows Prog. (Prereq: COP3530)

* Pre-requisites:

- COP2047: MAC1140 or higher level math course
- COT3100: Prereq: any MAC course and any COP course, Coreq:(COP2210 or COP2250 or EEL2280)
- ENC3249/ENC3213: UCC English requirements
- MAC2311: MAC1147
- MAD2104: MAC1105 or MGF1106

A direction line indicates a prerequisite. The course above must be completed before the course below can be taken.

A diamond indicates a co-requisite. The course closer to the diamond may be taken at the same time as the co-requisite. The co requisite is a prerequisite for any course that requires the course closer to the diamond.