

New Course Proposal

Submit completed form electronically

1. **Course prefix and course number:** CS - 357
2. **Course title:** Object Oriented Programming
3. **Abbreviated title for class schedule** (30 characters or less): OOP
4. **Credit hours:** 4
(note: if credits are variable, list range of credits (e.g. 1-8 credits))
5. **Catalog description:** The fundamental concepts of object-oriented programming languages, including data abstraction and typing, class inheritance, polymorphism and generic types, prototypes and delegation. Graphical User Interface Development. Introduction to UML and Design Patterns. To illustrate these issues, programming assignments in languages such as Java, C++, Python, etc. will be given.
6. **Prerequisites (to add each additional prerequisite, start a new line):**
(See attached Note for samples)
 - A. (course prefix, (space) and number) CS 257 with a C or better
 - B. (course prefix, (space) and number) or or
 - C. (course prefix, (space) and number) or or
7. **Co-requisites (including labs, if any):**
 - A. (course prefix, (space) and number) or or
8. **Major/Class restrictions:** Please indicate any class or major restrictions:
9. **Is course repeatable?** Yes No ☒ If Yes, list maximum credits:
10. **Labs requirements:** If course includes a lab: # of hours lecture: ; # of hours lab:
11. **Fees:** List any course fees:
12. **Grade Mode:** Graded only: Pass/No Pass only: Option: ☒
13. **CIP Code:** Six-digit CIP code (check with your Division Director): [11.0201](#)
14. **Special qualifications; Is course proposed for (yes/no):**
 - A. University Studies? If yes, list Strand(s)
 - B. Honors?

15. **Cross-listing:** List any cross-listing (and please complete the Cross-list proposal form at <https://inside.sou.edu/provost/curriculum.html>): and
and and

16. **Strategic justification for proposed course:**

A. **Rationale:** What is the overall strategic rationale for offering this course? *Preparing students to CS-469 (System Analysis) and CS-360 (Database), and improve and simplify our current prerequisite matrix.*

We are working to improve our programming course sequence and prepare students to learn how to program more agnostic (Focus on thinking instead of language). In this sense, our programming sequence is focusing on Problem Solving and Computational Thinking approaches.

To be more specific, currently, we are teaching two courses to cover two different programming languages, and our new approach is to focus on teaching two different paradigms instead, and languages be only their translations. Therefore, we are updating the current pair CS-257/CS-367 to become CS-257/CS-357.

We are updating our CS-257 to focus on imperative programming and CS-357 (this new course) will focus on Object Oriented Programming paradigm. As consequence, we will retire our current of CS-367 (C & Unix), because its content is being split between CS-257 and this new course.

Therefore, a specific course to cover Object-Oriented Programming would give to the students to see this paradigm in different languages and reinforce their programming skills in different languages. Furthermore, we are examining our prerequisite structure to remove unnecessary roadblocks for students.

We highlight that, we are not specifying languages in our descriptions, but we intend to use C and C++ in CS-257 and C++, Java, Python, etc in CS-357, as a language tool to train student skills.

Finally, these changes we are doing creating this course and updating CS-257 will help with better consistency and alignment with our ASOT-CS and MTM-CS and support transfer students.

B. **Alignment:**

1. How does this course align with the unit's mission plan? *Our new course aims to providing background and foundations of Object-Oriented Programming paradigm to our student. This sense we intend to create an environment to support creativity, critical thinking, discovering, and cultivate bold ideas and actions.*

2. How does the course fit into the rest of the unit's curriculum? *Offering this course, we will prepare students to System Analysis and Databases.*

C. **Enrollment:** What is the new course's estimated enrollment each time it is offered over a three-year period? Year 1 22 - 26 ; Year 2 22 - 26 ; Year 3 28 - 32

D. **Resource evaluation:** What resources – faculty, equipment, lab space, etc. -- will be needed to offer this course and how will those resources be obtained?

1. *Faculty:*

a. Who will teach the course? CS

- b. Evaluate unit's faculty availability and/or needs and the impact on other teaching obligations. *Since we are exchanging a course by another one, there is no change.*
- c. If additional faculty members are needed, how will that need be met? *TxT*

2. **Facilities:** Cite any additional need for classrooms, equipment or lab space; explain how that need(s) will be met. *This course will need to be taught in a computer lab to meet learning objectives.*

3. **Other:**

- a. Are Hannon Library resources sufficient to meet the needs of this course? *Yes*
- b. Are any other resources needed to support this course? *No*
If so, please explain how they will be obtained. *-*

E. External impact:

1. What is the expected effect of this course on existing programs elsewhere in the university? *Minimal if any.*

NOTE: Please document your contact with other academic programs which may be affected by this new course and the response you received.

2. Will any of your prerequisites affect other academic programs? *There is no impact in other programs. CS students are already taking CS 257 and CS 367, and this new course will replace and change the sequence of some programming topics, but no impact is expected.*

NOTE: Please document your contact with other academic programs which may be affected by this new course and the response you received.

17. Syllabus (condensed)

*(Attach an accompanying, condensed syllabus, which should include the following items. Schedules and similar details are **not** required.)*

A. Course description (same as Catalog description, above)

The fundamental concepts of object-oriented programming languages, including data abstraction and typing, class inheritance, polymorphism and generic types, prototypes and delegation. Graphical User Interface Development. Introduction to UML and Design Patterns. To illustrate these issues, programming assignments in languages such as Java, C++, Python, etc. will be given.

B. Learning objectives of the course

The main goal of this course is that students obtain those skills in Object-Oriented Programming paradigm and an introduction to design patterns. Upon the successful completion of this course students will be able to:

Understand the principles of object-oriented programming in a higher-level programming language, such as C++, Java, Python, etc.;
Analyze a problem statement to develop a mental model of objects necessary to create a software architecture;

Utilize object-oriented programming to frame software architectures, with care towards separation of concerns and abstraction;
Gain skills in designing, and programming software for reuse of code;
Establish development methods in object-oriented programming to qualify students for teaching the language in other settings.
Apply object-oriented programming or non-oriented techniques to solve bigger computing problems.

C. Required texts

*Bjarne Stroustrup. **The C++ Programming Language**. Addison-Wesley Professional; 4th edition. May 19, 2013.*

*Graham Lee. **Modern Programming: Object Oriented Programming and Best Practices: Deconstruct object-oriented programming and use it with other programming paradigms to build applications**. June 28, 2019. Packt Publishing.*

D. Course format

The course is devoted to problem solving techniques using Object-Oriented Programming paradigm. The exercises and tests require problem analysis to find out which design fits best to solve a problem. Therefore, the course format is based on interactive lectures in a computer lab, where instructors provides not only techniques, demonstrations and content, but also, exercises to train and challenge students, in a sense to develop their problem solving reasoning.

E. Other – any other relevant materials needed to explain the goals and teaching methods of this course.

Approvals:

Signature of Division Director

Date

4/29/16