Faculty Senate Course Form

Effective Date: **Fall 2025** Submission Date: 10/25/2024

Department: Mathematics and Physics College of: **Arts & Sciences**

Contact Person: Bobby Winters Prefix: **Dr**

Create New, Revise, Inactivate, or Reactivate: **New** Course #: **CS 300**

Course Form:
- Used to create new course numbers or new prefixes.
- Used to change Name, Grading, Hours, Description, Reactivate
- Used to inactivate a course from the current catalog. Courses are never deleted. They are made inactive and can be legislated to become active again.

1. Purpose/Justification for the Changes:

The purpose of this course is to provide an intermediate computer programming experience taught by faculty of the Computer Science program. It is meant to substitute for DSIS 240 (formerly CIS 240) in the Computer Science program. This has been approved by the Dean of the Kelce College of Business. (Please see the attached.)

2. Is this related to, and/or affect, any other department/college/unit curricula or programs at Pittsburg State University? ***If "Yes", please provide an explanation. Provide documentation of any discussions (e.g. copies of emails, memos, etc.) that have occurred.***

(●) Yes ( ) No

3. Is this course to be considered for General Education?
If "yes" this requirement will need approval of the General Education Committee after the revisions have been approved by Faculty Senate. The General Education Course Approval form will also need to be submitted.

( ) Yes (●) No

4. Will this course be required of any education majors?
If "yes," this requirement will need approval of the Council for Teacher Education before upload to " College Curriculum Legislation" in SharePoint.

( ) Yes (●) No

5. Will additional resources or costs be required?

( ) Yes (●) No

If so, what will be needed?

6. Will any additional course fees be required (e.g. equipment, clothing, travel, licensing, etc.)?
If "yes," complete the Course Fee Form on the Faculty Senate website, it will need to gain approval of the President's Council.

○ Yes ◉ No

7. Objectives/Student Learning Outcomes for NEW courses only, as it will appear in the syllabus:
Attach with upload.

8. Assessment Strategies (e.g. exams, projects, university rubric, etc.), as it will appear in the syllabus:
Attach with upload.

Course Numbers cannot be changed, only created.

	Exsisting	New/Proposed
Title:		Computer Programming 2
Course Number:		**CS 300**
Credits:		**3**
Grading System:	**Select One**	**A-F, IN**
Pre/Co-Requisite(s):		CS 200, CIS 230, or consent of the instructor.
Course Description:		Based in object-oriented programming, the course covers intermediate concepts of programming using a high-level programming language. It covers algorithm development, design concepts, pattern development, and application architecture. It is intended to be a second course in programming concepts .

Authorization Sign-Off

Checklist

- [x] Required fields completed.
- [x] Syllabus attached for new courses
- [x] Assignment Strategies Attached

-Approved: Department Chair/Director
Date: 10/25/2024 Signature, Chair/Director: [signature]

-Approved: College Curriculum Committee
Date: 12/2/24 Signature, Committee Chair: Christopher Childers

-Approved: Dean of College
Date: 12/2/24 Signature, Dean: Christopher Childers

-Approved: Council for Teacher Education (if applicable)
Date: _____ Signature, Council Chair: _____

-Approved: University Undergraduate Curriculum Committee
Date: _____ Signature, Committee Chair: _____

-Approved: Faculty Senate
Date: _____ Signature, Recorder Faculty Senate: _____

Originating Departments(s): After completing this form, please upload it to the SharePoint, within the appropriate College folder, "Preliminary Legislation", to allow for review and questions. Any modifications should be saved as "original file name.v2.docx" and uploaded as well.

Following final College Curriculum Committee approval, please apply the appropriate signatures, and send them to your College Administrator.

Outlook

Re: Introductory Computer Science Programming Courses

From Paul Grimes <pgrimes@pittstate.edu>
Date Fri 9/13/2024 7:42 AM
To Bobby Winters <bwinters@pittstate.edu>
Cc Tim Flood <tflood@pittstate.edu>; Alex Binder <abinder@pittstate.edu>; David Sikolia <dsikolia@pittstate.edu>

Professor Winters -

Thank you for your email concerning the proposal to create new programming courses for the Computer Science undergraduate degree program. Yes, the Kelce College of Business does not object to this proposal and supports the idea of new CS courses that will complement our existing DSIS programming courses. This support is endorsed by our DSIS faculty as stated during our meeting.

We look forward to finding additional ways in which we can work together to support our students and mutual interests in furthering the mission of our university.

All the best,

Paul

Paul W. Grimes, Dean
Kelce College of Business
Pittsburg State University
(620) 235-4590
https://orcid.org/0000-0002-3938-9696

From: Bobby Winters <bwinters@pittstate.edu>
Sent: Thursday, September 12, 2024 2:50 PM
To: Paul Grimes <pgrimes@pittstate.edu>
Cc: Tim Flood <tflood@pittstate.edu>; Alex Binder <abinder@pittstate.edu>; David Sikolia

<dsikolia@pittstate.edu>
Subject: Introductory Computer Science Programming Courses

Dean Grimes,
As per our meeting today, the Department of Mathematics and Physics is planning to legislate the attached courses. These are meant to be Computer science versions of CIS 230 Introductions to Programming and CIS 240 Intermediate Programming, respectively.

We are proposing these courses as a means to allow us to teach them ourselves as our staffing allows it so as to take the pressure off the fully subscribed courses that are taught by the Kelce College of Business. Our hope is that in the fullness of time, these courses might be taught in such a way as to provide synergy between our complementary programs and maximize both the university's resources and opportunities for our students.

Please let me know if you have any questions or concerns.

Bobby Winters
Professor of Mathematics
Associate Dean of the College of Arts and Sciences

Phone: 620-235-4788
Office: 200 Yates Hall

Student Learning Outcomes: CS 300, Computer Programming 2

These were developed by the Kansas Core Outcomes Group on Computer Science

Upon completion of this course, students will be able to:

- Explain and apply core object-oriented programs principles, including subclasses, encapsulation, inheritance, and abstraction.
- Develop robust programs by implementing error-handling techniques, including exception handling, in object-oriented programs.
- Enumerate the differences between imperative and object-oriented programming paradigms.
- Compose a class through design, implementation, and testing to meet behavioral requirements.
- Demonstrate knowledge and use of object-oriented programming, collection classes, and iterators and apply them effectively in problem-solving.
- Use structured problem-solving techniques to decompose complex problems and develop effective, efficient solutions through object-oriented programming principles.

CS 300, Computer Programming 2

1. **Outcome:** Explain and apply core object-oriented programs principles, including subclasses, encapsulation, inheritance, and abstraction.

Strategy: There will be a series of programming assignments in which these concepts will be required. Students will then write short reports regarding where they were used and what they are.

2. **Outcome:** Develop robust programs by implementing error-handling techniques, including exception handling, in object-oriented programs.

Strategy: Students will be given object-oriented programs to which they must add error-handling techniques.

3. **Outcome:** Enumerate the differences between imperative and object-oriented programming paradigms.

Strategy: This will be embedded as a question in the final exam.

4. **Outcome:** Compose a class through design, implementation, and testing to meet behavioral requirements.

Strategy: This will be done as a stand-alone programming assignment near the end of the course.

5. **Outcome:** Demonstrate knowledge and use of object-oriented programming, collection classes, and iterators and apply them effectively in problem-solving.

Strategy: This will be spread over a series of programming assignments distributed throughout the semester.

6. **Outcome:** Use structured problem-solving techniques to decompose complex problems and develop effective, efficient solutions through object-oriented programming principles.

Strategy: There will be a final individualized project that requires solving a problem through the use of object oriented programming.