

Faculty Senate Course Form

Effective Date: **Fall 2025**

Submission Date: 10/25/24

Department: Mathematics and Physics

College of: **Arts & Sciences**

Contact Person: Bobby Winters

Prefix: **Dr**

Create New, Revise, Inactivate, or Reactivate: **New**

Course #: **CS 200**

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 introduction to computer programming taught by faculty of the Computer Science program. It is meant to substitute for DSIS 230 (formerly CIS 230) in the Computer Science program. This has been approved by the Dean of the Kelce College of Business.

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 **NO, the form defaults to YES**

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 **NO, the form defaults to YES**

5. Will additional resources or costs be required?

Yes No **NO, the form defaults to YES**

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.

	Existing	New/Proposed
Title:		Computer Programming 1
Course Number:		CS 200
Credits:		3
Grading System:	Select One	A-F, IN
Pre/Co-Requisite(s):		MATH 019 Intermediate Algebra or MATH 110 College Algebra with Review or MATH 113 College Algebra.
Course Description:		This course will teach the basic skills of computer programming in a contemporary computer language.

Authorization Sign-Off

Checklist

- | | |
|-------------------------------------|-----------------------------------|
| <input checked="" type="checkbox"/> | Required fields completed. |
| <input checked="" type="checkbox"/> | Syllabus attached for new courses |
| <input checked="" type="checkbox"/> | Assignment Strategies Attached |

-Approved: Department Chair/Director

Date: 10/25/2024

Signature, Chair/Director: _____



-Approved: College Curriculum Committee

Date: 12/2/24

Signature, Committee Chair: _____



-Approved: Dean of College

Date: 12/2/24

Signature, Dean: _____



-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.

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 200, Computer Programming 1

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

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

- Explain key programming concepts such as variables, data types, control structures, functions, arrays.
- Design and implement basic algorithms for solving common computational problems. Students will be able to write, test, and debug programs using appropriate syntax and best practices in a programming language (e.g., Python, Java, C++).
- Demonstrate proficiency in using conditional statements (if-else) and loops (for, while) to control program flow.
- Create and use functions to organize and modularize code, applying principles of reusability and abstraction.
- Write programs that handle basic user input and display output, including file I/O operations.
- Demonstrate knowledge of fundamental data structures, such as arrays and/or lists, and apply them effectively in problem-solving.
- Use structured problem-solving techniques to decompose complex problems and develop effective, efficient solutions through programming.

CS 200, Computer Programming 1

1. Outcome: Explain key programming concepts such as variables, data types, control structures, functions, arrays

Strategies:

- Quizzes
- Tests

2. Outcome: Design and implement basic algorithms for solving common computational problems. Students will be able to write, test, and debug programs using appropriate syntax and best practices in a programming language (e.g., Python, Java, C++).

Strategy: Student will be assigned to write a program to solve a computational problem. It will be graded according to a rubric that addresses the issues indicated.

3. Outcome: Demonstrate proficiency in using conditional statements (if-else) and loops (for, while) to control program flow.

Strategy: Student will be assigned to write a program to implement a choice or decision. It will be graded according to a rubric that addresses the issues indicated.

4. Outcome: Create and use functions to organize and modularize code, applying principles of reusability and abstraction.

Strategy: Student will be assigned to write a complex program which requires organization. It will be graded according to a rubric that addresses issues of broad readability for the program in terms of later refactoring and debugging.

5. Outcome: Write programs that handle basic user input and display output, including file I/O operations.

Strategy: Student will be assigned to write a program that requires input and output. It will be graded according to a rubric that addresses the issues indicated.

6. **Outcome:** Demonstrate knowledge of fundamental data structures, such as arrays and/or lists, and apply them effectively in problem-solving.

Strategies:

- Quizzes
- Tests

7. Outcome: Use structured problem-solving techniques to decompose complex problems and develop effective, efficient solutions through programming.

Strategy: Student will be assigned a larger final project that includes the many of the previous goals and requires integration of knowledge through problem-solving. It will be graded according to a rubric that addresses the issues indicated.