

Faculty Senate Course Form

Effective Date: **Fall 2026**

Submission Date: 9/5/2025

Department: Mathematics and Physics

College of: **Arts & Sciences**

Contact Person: Bobby Winters

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

Course #: **CS 235**

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 New course or Changes to existing:

The need for an course to introduce students to the basic architecture of chips/processors and of programming chips/processors.

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?

PSU Faculty Senate 25-26

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:		
Course Number:		CS 235
Credits:		3
Grading System:	Select One	A-F, IN
Pre/Co-Requisite(s):		Pre-requisite CS-200 Computer Programming 1 or DSIS-230 Introduction to Programming
Course Description:		The course is to be taught at an introductory level. The course will introduce the student to the basics of the architecture of a particular chip or processor. The student will learn to program the chip or processor at a very basic level. The student will be taught about binary and hexadecimal numbers. The course assumes that the student is comfortable working with a computer.

Authorization Sign-Off

Checklist: Check once verified.

✓
✓
✓

Required fields completed.

Syllabus attached for new courses

Assignment Strategies Attached

-Approved: Department Chair/Director

Date: 9/5/25

Signature, Chair/Director: _____



-Approved: College Curriculum Committee

Date: 10/7/25

Signature, Committee Chair: _____



-Approved: Dean of College

Date: 10/7/25

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.

CS 235 Introduction to Low-Level Programming

Student Learning Outcomes

1. Working with an appropriate Integrated Development Environment.
2. Basic chip/processor architecture, e.g. general purpose registers, special registers, arithmetic-logic processing unit, data bus, etc.
3. Working with basic operations in assembly language to:
 - a. Move bytes from one general purpose register to another.
 - b. Move bytes between general purpose registers and RAM using pointers.
 - c. Move bytes from program memory to general purpose registers using pointers.
4. Working with basic arithmetic operations in assembly language for:
 - a. Addition
 - b. Subtraction
 - c. Multiplication
 - d. Comparison
5. Introduction to branching operations such as
 - a. Jump
 - b. Various conditional branches
 - c. The calling of subroutines
6. Working with various quasi-peripheral devices such as but not limited to:
 - a. Shift-Registers
 - b. Analog to Digital Converters
7. Introduction to good practices in organizing code such as:
 - a. Creation of Modular Code
 - b. Eschewing of “Magic Numbers”

Assessment Strategies for Introduction to Low-Level Programming

1. Quizzes for concepts and vocabulary. This will be used to assess knowledge of architectural concepts, knowledge of hexadecimal numbers, etc.
2. Written assignments to document code. This will be done to test the understanding of the functioning of code.
3. Modifying code examples to achieve a new purpose. This will further test the understanding of the functioning of code.
4. Creation of macros in order to learn abstraction. This tests the ability to abstract from particular examples.
5. Team class projects in programming. This tests the ability to integrate knowledge of the material.