



## Course Syllabus

### **COSC 1437 – Programming Fundamentals II**

*Revision Date: 08/22/2016*

**Catalog Description:** This course focuses on the object-oriented programming paradigm, emphasizing the definition and use of classes along with fundamentals of object-oriented design. The course includes basic analysis of algorithms, searching and sorting techniques, and an introduction to software engineering processes. Students will apply techniques for testing and debugging software. (This course is included in the Field of Study Curriculum for Computer Science.)

**Lecture Hours = 3, Lab Hours = 3**

**Prerequisites:** COSC 1436– Programming Fundamentals I

**Semester Credit Hours: 4**

**Lecture Hours per Week: 3**

**Lab Hours per Week: 3**

**Contact Hours per Semester: 96**

**State Approval Code: 11.0201.55 07**

#### **Core Components and Related College Student Learning Outcomes**

This course counts as part of the academic requirements of the Panola College Core Curriculum and an Associate of Arts or Associate of Science degree.  Yes  No: If no, skip to Instructional Goals

#### **Instructional Goals and Purpose**

Panola College's instructional goals include 1) creating an academic atmosphere in which students may develop their intellects and skills and 2) providing courses so students may receive a certificate/an associate degree or transfer to a senior institution that offers baccalaureate degrees.

The purpose of this course is to fulfill one of the academic requirements in the Computer Science field of study for students planning to transfer to a senior institution that offers baccalaureate degree in Computer Science and to introduce students to develop computer programs.

#### **Learning Outcomes**

Upon successful completion of this course, students will:

1. Identify and explain a programming development lifecycle, including planning, analysis, design, development, and maintenance.
2. Demonstrate a basic understanding of object-oriented programming by using structures and classes in software projects.
3. Use object-oriented programming techniques to develop executable programs that include elements such as inheritance and polymorphism.
4. Document and format code in a consistent manner.
5. Apply basic searching and sorting algorithms in software design.
6. Apply single- and multi-dimensional arrays in software.
7. Use a symbolic debugger to find and fix runtime and logical errors in software.
8. Demonstrate a basic understanding of programming methodologies, including object oriented, structured, and procedural programming.
9. Describe the phases of program translation from source code to executable code

## Course Content

Students in all sections of this course will learn the following content:

1. Create programs using functions.
2. Write modules that perform basic manipulations with one-dimensional arrays
3. Code statements iterating over sequences
4. Understand variable scope and be able to use variables appropriately in programs.
5. Create software using problem solving skills to design, code and test programs.
6. Demonstrate the use of debugging functions available in software development tools.
7. Run, test and debug programs.
8. Students will submit computer programs for each learning module of the course. Each program must demonstrate comprehensive knowledge of the learning module represented.
9. Students will assemble a portfolio of programming projects to be submitted at the end of the semester.
10. Students will conduct an interactive presentation of a comprehensive programming project subject to peer and instructor evaluation.
11. Students will submit computer programs for each learning module of the course. Each program must demonstrate comprehensive knowledge of the learning module represented. Students will assemble a portfolio of programming projects to be submitted at the end of the semester.

## Methods of Instruction/Course Format/Delivery

The method of instruction is implemented with student use of a text book, **An Introduction to C++ Programming** by Diane Zak. The book uses a pedagogy that works well with our students.

- Each chapter introduces new material subdivided into short sections.
  - Sections are a “divide and conquer” approach to presenting information that might be overwhelming if presented in a lengthier form.
    - Each section briefly, but thoroughly, explains a single topic thus simplifying comprehension.
  - Understanding of each section is reinforced with opportunities to complete short activities, projects, or quizzes as they work through each chapter.
  - End of chapter comprehensive projects are provided to assure the students that they are ready for the next chapter of the text.
- The chapters follow a pattern of building programming knowledge by beginning with an introduction to basic computer concepts followed by basic programming concepts.
- After the computer and programming concepts are taught, students are introduced to logic concepts and how programs may be designed using algorithms or flowcharts before program code is written.
- The course formats include traditional (face-to-face), hybrid, and online. Students in classes of any format will have access to this course via Canvas. Resources provided through Canvas include: email allowing communication with all class members and the instructor, project data files, installation file for compiler installation, overview of the course, a manual for compiler installation and use, exams, a variety of exercises including programming projects.

## Assessment

All assignments will be submitted through Canvas. After the assignment has been graded, the student will be able to view his or her grade by returning to the assignment and clicking the “View Scores” button or “My Grades” link. All exams will either be hands-on application or tests.

If a student is having problems with a particular programming project, they are asked to email the source code to me and they receive a response indicating what is causing the problem.

## Course Grade

The grade for this course will be based on the following items assigned during the semester and used to calculate the student’s final grade:

### **ASSIGNMENTS**

Students work through each required chapter in sequence. At the end of each chapter, they complete a programming project demonstrating their knowledge of the programming concepts.

Program source code is submitted according to the schedule presented in Canvas using an online drop box in the “Assignments” link of Canvas.

### **EXAMS**

There will be two assessments to verify that you have the comprehensive knowledge required to produce your portfolio. You will demonstrate this knowledge by conducting an interactive presentation of a comprehensive programming project subject to peer and instructor evaluation.

### **Grade Calculation**

The grading calculation course is as follows:

- Assignments – 30%
- Comprehensive programming project – 30%
- Exams – 40%

All grades including a mid-semester and final grade will be posted to My Grades in Canvas.

## Texts, Materials, and Supplies

- Required: **An Introduction to Programming with C++, 8th Edition** by Diane Zak, 2016, Cengage
- Access to a computer and the Internet.
- Student data files and Dev-C++ are provided with the textbook.
- Method of saving source code: USB Drive, Cloud Drive, etc.

## Required Readings:

- **An Introduction to Programming with C++, 8th Edition by Zak**

## Recommended Readings:

- Programming and c++ websites listed in Canvas.
- Try doing a Google search on topics, terms, functions, etc. that you are not understanding from the text material. You will find a good explanation on the Internet is your patient is your search. supplemental reading will be provided as handouts or an Internet link.

## Other:

- Following link: <http://www.panola.edu/elearning/testing.html>
- If any student in this class has special classroom or testing needs because of a physical learning or emotional condition, please contact the ADA Student Coordinator in Support Services located in the Administration Building or go to <http://www.panola.edu/student-success/disability-support-services/> for more information.
- Withdrawing from a course is the student’s responsibility. Students who do not attend class and who do not withdraw will receive the grade earned for the course.