

COSC 218-COMPUTER PROGRAMMING FUNDAMENTALS II 3 CREDITS

SYLLABUS

CATALOG DESCRIPTION

Intermediate level object-oriented programming course with emphasis on the underpinnings of object-oriented design. The fundamental tenets of object-oriented program such as inheritance, polymorphism, pointers, templates, and dynamic memory management are covered in depth. The Standard Template Library is presented as an introduction to data structures.

Prerequisites: COSC 118

Semester Offered: Spring

Common Student Learning Outcomes

Upon successful completion of San Juan College programs and degrees, the student will demonstrate competency in...

BROAD AND SPECIALIZED LEARNING

Students will actively and independently acquire, apply, and adapt skills and knowledge with an awareness of global contexts.

CRITICAL THINKING

Students will think analytically and creatively to explore ideas, make connections, draw conclusions and solve problems.

CULTURAL AND CIVIC ENGAGEMENT

Students will act purposefully, reflectively, and ethically in diverse and complex environments.

EFFECTIVE COMMUNICATION

Students will exchange ideas and information with clarity in multiple contexts.

INFORMATION LITERACY

Students will be able to recognize when information is needed and have the ability to locate, evaluate, and use it effectively.

INTEGRATING TECHNOLOGIES

Students will demonstrate fluency in the application and use of technologies in multiple contexts.

Student work from this class may be randomly selected and used anonymously for assessment of course, program, and/or institutional learning outcomes. For more information, please refer to the Dean of the appropriate School.

Course Learning Outcomes

Upon successful completion of the course, the student will be able to...

1. To improve and refine students' problem solving skills and use of abstraction.
2. To equip students with a thorough understanding of the C++ language and syntax.
3. To provide students with hands-on programming experience in object-oriented programming.
4. To provide students with the skills and experience to create their own designed classes.

Specific Learning Outcomes

Upon successful completion of the course, the student will be able to ...

1. Become proficient in object-oriented design
2. Thoroughly understand class design issues (public versus private; choosing members)
3. Thoroughly understand the role of constructors, accessors, modifiers and how to implement them
4. Become proficient in writing class definitions
5. Become proficient implementing classes including data members and member functions given the class definition
6. Demonstrate encapsulation and how to hide implementation inside C++ classes
7. Describe the concepts and syntax related to C++ implementation of inheritance
8. Apply object-oriented design heuristics for inheritance
9. Derive new classes from old ones employing inheritance
10. Demonstrate the need for copy constructors and destructors
11. Override member functions adding new features to derived classes
12. Understand and implement polymorphism
13. Implement class templates
14. Implement function templates
15. Pass class names to container classes
16. Implement generic container classes that store collections of any class of elements
17. Study, design, and implement operator overloading using logical, math, and assignment operators
18. Present and use advanced $O(n^2)$ sorting algorithms
19. Show how standard iterator objects are used to implement the iterator pattern
20. Use the iterator pattern to traverse a collection of elements without revealing the underlying implementation
21. Develop an understanding of pointers
22. Implement pointer objects that store addresses of other objects
23. Understand, use, and implement `char*` objects
24. Show benefits of dynamic memory allocation techniques
25. Implement dynamic memory allocation using `new`, `delete`, and classes
26. Understand and implement a collection stored as a linked list
27. Study recursion and recursive problem solving
28. Plan and implement recursion

COURSE CHALLENGE PROCEDURES

This course may be challenged by taking and passing all the semester tests and the final exam. A grade of 70% is required to successfully challenge this course.