

Minnesota State College Southeast

COMC 2742: Java/C++/C# Programming II

A. COURSE DESCRIPTION

Credits: 3

Lecture Hours/Week: 2

Lab Hours/Week: 2

OJT Hours/Week: *.*

Prerequisites: None

Corequisites: None

MnTC Goals: None

This course covers object oriented programming concepts using the Java, C#, and C++ languages. Topics include: class declarations, class methods and attributes, creating and using objects, constructors and destructors, function overloading, passing objects as function arguments, class inheritance, memory allocation, and object associations/aggregate objects, exception handling, exception classes, MVC architecture, user interface I/O, IoT sensor and actuator I/O, XML and/or JSON deserialization, source control, and unit tests. (Prerequisite: COMC1733 or instructor permission) (3 credits: 2 lecture/1 lab)

B. COURSE EFFECTIVE DATES: 05/21/1998 - Present

C. OUTLINE OF MAJOR CONTENT AREAS

D. LEARNING OUTCOMES (General)

1. Describe class concepts
2. Describe encapsulation
3. Perform data member declaration
4. Perform member function declaration
5. Create simple constructor function
6. Create parameterized constructor functions
7. Create destructor functions
8. Describe dynamic memory allocation
9. Create dynamic memory allocation programs
10. Describe aggregate object constructor concepts
11. Describe object-reference parameter passing concepts
12. Create object-reference parameter passing programs
13. Create object-pointer programs
14. Describe multiple-source file project concepts
15. Create aggregate object constructor functions
16. Describe object comparison concepts
17. Create object comparison functions
18. Describe object copy concepts
19. Create object copy functions
20. Create same-class operation function
21. Override base class functions
22. Debug null-reference errors
23. Describe inheritance
24. Describe base class constructor concepts
25. Create base classes
26. Create derived classes
27. Create class inheritance programs
28. Describe polymorphism
29. Describe virtual function concepts
30. Create model/view programs
31. Create XML and/or JSON deserialization functions
32. Create exception handling programs
33. Utilize exception classes
34. Perform Swing user interface I/O
35. Perform file I/O
36. Read IoT sensor input
37. Perform IoT actuator output
38. Perform unit tests
39. Utilize Git source control
40. Display professional attitude

E. Minnesota Transfer Curriculum Goal Area(s) and Competencies

None

F. LEARNER OUTCOMES ASSESSMENT

As noted on course syllabus

G. SPECIAL INFORMATION

None noted