South Central College

COMP 2312 Software Development

Common Course Outline

Course Information

Description Software Development covers common programming techniques used in writing applications as well as demonstrating how to use the current leading Integrated Development Environment. Topics include object-oriented programming, control statements, database programming, and producing web-based applications. The capstone project for this course involves creating an application for a real-life business problem. (Prerequisite: Successful completion of COMP 1130 Programming Fundamentals with a C or higher, or instructor permission if the student has a working knowledge of at least one programming language.)

Total Credits 4

Total Hours 64

Types of Instruction

Instruction Type	Credits/Hours
Lecture	4/64

Pre/Corequisites

Successful completion of COMP 1130 Programming Fundamentals with a C or higher, or a working knowledge of at least one programming language.

Institutional Core Competencies

Communication - Students will be able to demonstrate appropriate and effective interactions with others to achieve their personal, academic, and professional objectives.

Critical and Creative Thinking - Students will be able to demonstrate purposeful thinking with the goal of using a creative process for developing and building upon ideas and/or the goal of using a critical process for the analyzing and evaluating of ideas.

Course Outcomes

1. Install and use an Integrated Development Environment (IDE).

Learning Objectives
Install IDE on your system.
Create and run a simple program using C#.

Describe why programming style is important in a team environment.

2. Use OOP (Object Oriented Programming) and UML (Unified Modeling Language) concepts to design a program.

Learning Objectives

Describe the concepts of OOP, including inheritance, polymorphism, encapsulation.
Draw a UML diagram to document a class.
Compare the differences between a flowchart and a UML diagram.
Utilize an algorithm and pseudocode to design programs.

3. Utilize various data types and operations in a program.

Learning Objectives

Write programs that utilize the different data types.
Use constants as part of your program design.
Convert String data into numbers.
Convert numeric data into Strings.
Demonstrate the different ways a number can be represented using binary, decimal, and hexadecimal.

4. Utilize methods for code reuse and code organization.

Learning Objectives

Utilize methods for greater code reuse.
Use parameters to send data to methods.
Write methods which return data.
Capture and use the data returned by a method using the assignment operator.

5. Use the built-in String methods.

Learning Objectives

Determine the length of a String.
Add multiple Strings together using the concatenation.
Utilize common String methods, for example, extraction, search, alphabetization.

6. Make programming decisions using the if, switch, and ternary constructs.

Learning Objectives

Make a yes/no decision using the if statement.
Use the ternary statement to make a decision.
Compare a value against multiple options with the switch statement.

7. Do repetitive tasks using the for, while, and do…while loops.

Learning Objectives

Repeat a set of statements a specific number of times using the for loop.
Loop indefinitely using the while and do…while loop statements.

8. Use built-in data structures to organize program data.

Learning Objectives

Utilize arrays, stacks, queues, linked lists, and sorting algorithms.
Describe the performance implications of various data structures.
Choose the right data structure.

9. Write your own custom classes.

Learning Objectives

Describe a Class and object using real-world examples.
Draw a UML diagram to show the contents of a Class.
Write a constructor that will initialize a Class into an object.
Demonstrate method overloading in a program.

10. Incorporate error handling for user-friendly programs.

Learning Objectives

Describe the types of errors that are encountered in programming.

Demonstrate effective debugging techniques using IDE debug tools.
Use structured error processing using the try, catch, and finally statements.

11. Explain general software development.

Learning Objectives

Describe Application Life Cycle Management, including software testing.
Read and translate application specifications into prototypes, code, and components.

12. Utilize databases.

Learning Objectives

Read and write data to a disk file.
Connect to a database.
Display data.

13. Use SQL (Structured Query Language) to communicate with databases.

Learning Objectives

Use the SELECT statement to extract specific information from a database.
Use the JOIN to extract information from several tables.
Use the UPDATE statement to update a database.
Write and use a stored procedure to run protected queries in a database program.
Delete a database table using SQL.

14. Create a web application.

Learning Objectives

Demonstrate the ASP.NET MVC (Model-View-Controller) Web Application Development.
Create an illustration showing the relationship between the client and server.
Create a dynamic web page using C#.
Publish the web page out to a live server on the Web.

SCC Accessibility Statement

Disability Services provides accommodations and other supports to students with permanent and temporary disabilities that affect their SCC experience. Disabilities may include mental health (anxiety, depression, PTSD), ADHD, learning disabilities, chronic health conditions (migraine, fibromyalgia), sensory disabilities, and temporary disabilities (broken arm, surgery). Common accommodations are extended test time, private room for testing, audiobooks, and sign language interpreter.

Contact us: Faribault A116 (507) 332-5847. North Mankato E112 (507) 389-7222. ds@southcentral.edu
www.southcentral.edu/disability