BAKER COLLEGE
STUDENT LEARNING OUTCOMES

CS 3110 C# Programming
3 Semester Hours

Student Learning Outcomes and Enabling Objectives

1. Use problem-solving techniques related to the C# programming language.
    a. Develop a technique to solve problems and implement their solutions using C#.
    b. Write object-oriented C# programs to solve problems.
    c. Use an integrated development environment to build C# programs.
2. Revisit the elements of object-oriented programming languages using C#.
    a. Review primitive data types (e.g., integers, floats, doubles, characters, etc.) and their operators.
    b. Review arrays, strings, and Classes including objects, properties, and methods.
    c. Review conditionals, loops, pointers, and reference types.
    d. Review input/output.
3. Analyze Functions.
    a. Review parameters, passing by value, passing by reference, using pointers.
    b. Review typed and untyped functions.
    c. Review scope and local variables.
    d. Review passing arrays, structures, objects.
4. Analyze Classes.
    a. Use constructors and object initialization.
    b. Use single and multiple inheritance when defining classes.
    c. Use overriding and overloaded functions.
    d. Use visibility modifiers to control access.
5. Examine the use of events.
    a. Understand throwing, catching, and blocking exceptions.
    b. Create and use custom exceptions.
6. Investigate the use of recursion.
    a. Explain direct and indirect recursion.
    b. Discuss infinite recursion.
    c. Compare and contrast the advantages/disadvantages of recursion compared to iteration.
7. Develop working object-oriented programs using the C# programming language.
    a. Create Classes using the C# programming language.
        i. Encapsulate object data and behavior.
        ii. Create static and non-static Class members.
        iii. Use the **this** pointer.
        iv. Use polymorphism.

    b. Create and manipulate an array in the C# programming language.
        i. Use the **for** and **foreach** loop control structures.
        ii. Create and manipulate multidimensional arrays.
        iii. Use the **ArrayList** class.
    c. Create documentation for working object-oriented C# programs.
8. Investigate the use of multithreading.
    a. Explain how to create and manage threads.
    b. Develop asynchronous solutions.
9. Work as part of a team to design, create, test, debug, and document an object-oriented C# program.
    a. Discuss the importance of teams in the programming field.
    b. Create the documentation for a working C# program.
    c. Develop a working object-oriented C# program as a team.

Big Ideas and Essential Questions

Big Ideas

- Object-Oriented Programming Core Principles
- Problem solving and program development using the C# language
- Teamwork

Essential Questions

1. How do programmers solve problems?
2. What is the C# programming language?
3. Why do programmers use the C# programming language?
4. What are the elements of the C# programming language?
5. What facilities does C# provide to support object-oriented programming?
6. After solving a problem, how do programmers implement the solution as an object-oriented C# program?
7. How do programmers build Classes using the C# language?
8. How do programmers utilize procedural code within C# programs?
9. Why is teamwork so important in the programming field?

These SLOs are approved for experiential credit.

Effective: Spring 2020