

 COMPUTER SCIENCE
 DEPARTMENT

CS 110 - STUDENT LEARNING OUTCOMES

**Apply secure coding techniques to object‐oriented programming solutions.**

* Below Expectations: Describes secure coding techniques of an object-oriented program, such as public versus private members, data integrity, and data typing.
* Meets Expectations: Applies secure coding techniques to an object-oriented program.
* Exceeds Expectations: Devises a fully secure object-oriented program.

**Apply the program development process to problems that are solved using fundamental programming constructs and predefined data structures.**

* Below Expectations: Summarizes the phases of the program development cycle.
* Meets Expectations: With guidance during the design phase, produces working code and performs some testing.
* Exceeds Expectations: Develops a working program solution by implementing design, coding, and testing that includes error checking.

**Choose professional behavior in response to ethical issues inherent in computing.**

* Below Expectations: Explains the concepts of intellectual property, plagiarism, and software piracy.
* Meets Expectations: Chooses to respond professionally to ethical issues in computing, such as intellectual property, plagiarism, and software piracy.
* Exceeds Expectations: Values and respects intellectual property, and chooses to act professionally.

**Compare and contrast the primitive data types of a programming language; describe how each is stored in memory; and identify the criteria for selection.**

* Below Expectations: Names the built-in data types of the programming language.
* Meets Expectations: Differentiates among the built-in data types and explain when it is appropriate to choose one over another.
* Exceeds Expectations: Consistently produces programming solutions with the correct data types implemented.

**Decompose a program into subtasks and use parameter passing to exchange information between the subparts.**

* Below Expectations: With guidance translates a problem into a programming solution with subtasks.
* Meets Expectations: With guidance for program analysis and design, decomposes a problem into program components that share data.
* Exceeds Expectations: Independently analyzes a problem, formulates a design strategy, and decomposes a problem into program components that share data.

**Describe the language translation phases of compiling, interpreting, linking and executing, and differentiate the error conditions associated with each phase.**

* Below Expectations: Defines the programming language terms of compiling, interpreting, linking, executing, and error conditions.
* Meets Expectations: Describes the programming language translation phases of compiling, interpreting, linking, and executing, and error conditions
* Exceeds Expectations: Compares the programming language translation phases of compiling, interpreting, linking, and executing, and distinguishes the error conditions associated with each.

**Differentiate between the object-oriented, structured, and functional programming methodologies.**

* Below Expectations: Recognizes the differences and similarities of the object-oriented, structured, and functional programming methodologies.
* Meets Expectations: Differentiates between the object-oriented, structured, and functional programming methodologies.
* Exceeds Expectations: Compares and contrasts the three prominent methodologies of object-oriented, structured, and functional programming.

**Produce algorithms for solving simple problems and trace the execution of computer programs.**

* Below Expectations: Defines the steps necessary to solve a programming problem.
* Meets Expectations: Produces a working programming solution for a given algorithm.
* Exceeds Expectations: Develops a generic solution for an algorithm that can be used to solve a range of related problems.

**Write and build a working computer program with command line tools as well as an IDE.**

* Below Expectations: With guidance writes and builds a working computer program with command line tools as well as an IDE.
* Meets Expectations: Independently writes and builds a working computer program with command line tools as well as an IDE.
* Exceeds Expectations: Independently writes and builds a working computer program with command line tools from more than one Operating System as well as two or more different IDEs.

**Identify and describe the function of subsystems commonly used in contemporary computer systems.**

* Below Expectations: Identifies the subsystems commonly used in a contemporary computer system
* Meets Expectations: Identifies the subsystems and describes the function of each one.
* Exceeds Expectations: Explains the relationship of each subsystem to overall performance and logically identifies which subsystem is likely the origin of a system malfunction.

**Write a simple web page with a text editor.**

* Below Expectations: With guidance, creates a functioning web page.
* Meets Expectations: Independently creates a functioning web page.
* Exceeds Expectations: Creates a web page with dynamic content.