

COURSE CSE1120: STRUCTURED PROGRAMMING 2

Level: Introductory

Prerequisite: CSE1110: Structured Programming 1

Description: Students work with structured programming constructs by adding the selection and iteration program control flow mechanisms to their programming repertoire. They write structured algorithms and programs that use blocks to introduce an element of modularity into their programming practice.

Parameters: Access to appropriate computer equipment, software, to the Internet and support materials. Specifically, students must have access to a programming environment that encourages structured programming.

Supporting Courses: CSE1010: Computer Science 1, or any
Intermediate project course involving imperative programming

Outcomes: The student will:

- 1. demonstrate basic structured programming skills by writing algorithms to solve problems involving selection (decision making) and iteration (repetition)**
 - 1.1 analyze a problem and determine if it can be solved using an algorithm that employs an input/processing/output (IPO) approach
 - 1.2 determine if there is more than one IPO module present
 - 1.3 decompose the problem into its respective modules and identify the IPO components of each module
 - 1.4 identify what data is already available to the programmer and what must be inputted by the end user and organize into appropriate block or blocks using the appropriate program control structures
 - 1.5 identify the processing requirements and organize into appropriate blocks using the appropriate program control structures
 - 1.6 incorporate basic algorithmic idioms as required; e.g., accumulation, determining maximum or minimum values
 - 1.7 identify the output requirements and organize into appropriate blocks using the appropriate program control structures
 - 1.8 order components into an appropriate sequence where processing occurs only when all required data for a module is available and output occurs only after appropriate processing has occurred
 - 1.9 write the algorithm in an acceptable format; e.g., pseudocode, a structured chart
- 2. translate algorithms into source code, convert the source code into machine executable form, execute and debug, as required**
 - 2.1 maintain the IPO structure of the algorithm
 - 2.2 use appropriate internal and external documentation
 - 2.3 use appropriate basic (primitive) data types such as integers, real numbers, characters, strings, and Boolean values
 - 2.4 use appropriate variables and constants to hold data
 - 2.5 use literals and input commands, e.g., methods or operators, to provide data for processing

- 2.6 use assignment, arithmetical, relational, Boolean, and concatenation and interpolation operators, where appropriate, to process data
- 2.7 use basic processing idioms as required; e.g., accumulation, determining maximum or minimum values
- 2.8 use appropriate selection and iteration structures to avoid unconditional branching or exiting from the interior of a block including:
 - 2.8.1 nested conditional blocks
 - 2.8.2 nested iterative blocks
- 2.9 use output commands, e.g., methods or operators, to display processed data in an appropriately formatted form
- 3. analyze and compare the results of the program with the intent of the algorithm and modify, as required**
 - 3.1 use appropriate test data and debugging techniques to track and correct errors including:
 - 3.1.1 run-time errors; e.g., compiler, linker, syntax
 - 3.1.2 logic errors
- 4. demonstrate basic competencies**
 - 4.1 demonstrate fundamental skills to:
 - 4.1.1 communicate
 - 4.1.2 manage information
 - 4.1.3 use numbers
 - 4.1.4 think and solve problems
 - 4.2 demonstrate personal management skills to:
 - 4.2.1 demonstrate positive attitudes and behaviours
 - 4.2.2 be responsible
 - 4.2.3 be adaptable
 - 4.2.4 learn continuously
 - 4.2.5 work safely
 - 4.3 demonstrate teamwork skills to:
 - 4.3.1 work with others
 - 4.3.2 participate in projects and tasks
- 5. make personal connections to the cluster content and processes to inform possible pathway choices**
 - 5.1 complete/update a personal inventory; e.g., interests, values, beliefs, resources, prior learning and experiences
 - 5.2 create a connection between a personal inventory and occupational choices