**COURSE CSE1110: STRUCTURED PROGRAMMING 1**

| | |
|---|---|
| **Level:** | Introductory |
| **Prerequisite:** | None |
| **Description:** | Students are introduced to a general programming environment in which they write simple structured algorithms and programs that input, process and output data, use some of the more basic operators and data types, and follow a sequential flow of control. |
| **Parameters:** | Access to appropriate computer equipment, software, the Internet and support materials. Specifically, students must have access to a programming environment that encourages structured programming. |
| **Supporting Course:** | CSE1010: Computer Science 1 |
| **Outcomes:** | The student will: |

1. **demonstrate introductory structured programming skills by writing sequential algorithms to solve problems involving input, processing and output**
   1.1 describe the purpose and nature of an algorithm
   1.2 analyze a variety of simple algorithms and describe the task or tasks the algorithms are attempting to carry out
   1.3 analyze problems and determine if they can be solved using algorithms that employ an input/processing/output (IPO) approach
   1.4 decompose the problem into its input, processing and output components, and identify what data is already available to the program and what must be inputted
   1.5 sequence components appropriately so that processing occurs only when all required data is available and output occurs only after appropriate processing has occurred
   1.6 write the algorithm in an acceptable format; e.g., pseudocode, structured chart
   1.7 test the algorithm for failure as well as success with appropriate data
   1.8 revise the algorithm, as required
2. **translate algorithms into source code, convert the source code into machine executable form, execute and debug, as required**
   2.1 describe a typical programming development environment commenting on the role of the key components; e.g., the source code editor, code translator (compiler and/or interpreter), executor, debugger
   2.2 compare and contrast integrated development environments specifically developed for programming with user assembled collections of applications and system software; e.g., text processor for coding, command line compiler
   2.3 describe and demonstrate the use of key components in a programming development environment
   2.4 convert algorithms into a sequence of statements in an appropriate programming language being sure to:
   2.4.1 maintain the IPO structure of the algorithm
   2.4.2 use appropriate internal and external documentation
   2.4.3 use appropriate data types such as integers, real numbers, characters and strings
   2.4.4 use appropriate variables and constants to hold data

2.4.5 use literals and input commands, e.g., methods or operators, to provide data for processing

2.4.6 use assignment, arithmetical and concatenation and interpolation operators, where appropriate, to process data

2.4.7 use output commands; e.g., methods or operators, to display processed data

2.5 test the algorithm for failure or success with appropriate data

2.6 revise the algorithm, as required

**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