CPRG 216 - Object-Oriented Programming 1

Course Description:
This course explores the concept of computer programming languages as tools that can be used to provide IT business solutions. Topics include problem solving techniques, an overview of software design methodology, programming language terminology and an introduction to the differences between styles of languages. The course also includes an introduction to basic programming structures, and you'll create applications using an industry-standard programming language.

3 Credits

Time Guidelines:
The standard instructional time for this course is 75 hours.

Course Assessment:

Assignments	60%
Project(s)	40%
Total:	100%

Other Course Information:
Learner Engagement:

In order to be successful, the learner is expected to be engaged in learning activities for a total of 9 to 12 learning hours per course per week, which includes both in-class and out-of-class time.

SAIT Policies and Procedures:
For information on the SAIT Grading Scale, please visit policy AC 3.1.1 Grading Progression Procedure, found on the SAIT Academic Policies and Procedures page: https://www.sait.ca/about-sait/administration/policies-and-procedures

For information on SAIT Academic Policies, please visit: www.sait.ca/about-sait/administration/policies-and-procedures/academic-student

Course Learning Outcome(s):
1. Outline the evolution of computer programming.

Objectives:

1.1 Describe what computers do.

1.2 Define a programming language.

1.3 Illustrate the evolution of programming languages.

1.4 Describe how a computer interprets an algorithm.

1.5 Explain a program's flow of control.

1.6 Compare programming languages.

1.7 Describe the characteristics of a programming language.

2. Solve programming problems.

Objectives:

2.1 Describe the rules of syntax in programming.

2.2 Apply syntax rules to writing a program.

2.3 Explain conditional statements and branching.

2.4 Demonstrate the use of conditional statements and branching.

2.5 Describe repetition structures.

2.6 Demonstrate the use of repetition structures.

3. Explain code reuse.

Objectives:

3.1 Describe functions in programming.

3.2 Describe lambda abstraction.

3.3 Apply debugging.

3.4 Apply keyword arguments and default values.

3.5 Apply a variable number of arguments.

3.6 Apply scoping of variables.

3.7 Explain the concept of functions as objects.

3.8 Describe reusing methods.

3.9 Apply global variables.

4. Demonstrate classes and object-oriented programming principles in design projects.

Objectives:

4.1 Describe object-oriented programming.

4.2 Describe classes and their purpose.

4.3 Describe objects and their purpose.

4.4 Define attributes, class variables and instance variables.

4.5 Demonstrate how to create objects.

5. Demonstrate version control.

Objectives:

5.1 Define version control.

5.2 Explain the importance of version control.

5.3 Describe Git and Github.

5.4 Explain the basics of using Github in software projects.

5.5 Demonstrate how to use Github commands.