

Last Revised: October 2022

Last Reviewed: October 2022

**COURSE INFORMATION****Course Title:** Introduction to Computer Science & Programming II**Course Number:** CSCI 125**Credits:** 4**Total Weeks:** 14 (Fall, Spring)  
12 (Summer)      **Total Hours:** 65**Course Level:**     First Year       Second Year  
 New                       Revised Course  
 Replacement Course**Department:** Computer Science    **Department Head:** M. O'Connor**Former Course Code(s) and Number(s) (if applicable):**  
N/A**Pre-requisites (If there are no prerequisites, type NONE):**

- PREC 12 or equivalent and CSCI 120

**Co-requisite Statement (List if applicable or type NONE):** NONE**Precluded Courses:** N/A**COURSE DESCRIPTION**

This course is a thorough introduction to computer science and computer programming, suitable for students with some computer science background. It is designed for students who will major in computing science or a related program. Students will learn fundamental concepts and terminology of computer programming and acquire basic programming skills in the Java programming language. Topics covered are: primitive and abstract data types, elementary data structures, fundamental algorithms, algorithm analysis, basic object-oriented programming and software design, specification and program correctness, and historical aspects of computing science.

**LEARNING OUTCOMES**

Upon successful completion of the course, students will be able to:

- Cite some historical facts of computer science and programming.
- Demonstrate how to choose primitive data types appropriately.
- Know the difference between primitive data types and reference data types.
- Describe and apply techniques to document programs.
- Use appropriate tools for software development
- Develop, test, and evaluate programs
- Use good and defensive programming style
- Demonstrate how to use control structures appropriately.
- Demonstrate how to use text files for input and output.
- Know how to construct safe programs.
- Explain and use simple abstract data types such as list, stack and queue.
- Demonstrate how to design and use classes.
- Demonstrate how to use modeling tools such as UML, for object-oriented design.
- Develop object-oriented programming solutions
- Know data structures from the Java Collections API such as arrays, lists and dictionaries.
- Demonstrate how to use single and multi-dimensional arrays.
- Know and analyze fundamental searching and sorting algorithms.
- Explain and apply the concepts of Object-Oriented Programming.
- Define and use recursive algorithms for problem solving.

**INSTRUCTION AND GRADING**

Instructional (Contact) Hours:

Type	Duration
Lecture	39
Seminars/Tutorials	
Laboratory	26
Field Experience	
Other ( <i>specify</i> ):	
Total	65

**Grading System:** Letter Grades  Percentage  Pass/Fail  Satisfactory/Unsatisfactory  Other

**Specify passing grade:** 50%

**Evaluation Activities and Weighting** (total must equal 100%)

Assignments: 5% <i>Review Questions: 3%</i> <i>Review Quizzes: 2%</i>	Lab Work: 15%	Participation: % <i>Specify nature of participation:</i>	Projects: 15% <i>Midterm Project: 5%</i> <i>Final Project: 10%</i>
Quizzes/Test: %	Midterm Exams: 30% <i>Midterm 1: 15%</i> <i>Midterm 2: 15%</i>	Final Exam: 35%	Other: %

**TEXT(S) AND RESOURCE MATERIALS**

Provide a full reference for each text and/or resource material and include whether required/not required.

Tony Gaddis, Godfrey Muganda: Starting Out with Java: From Control Structures through Data Structures, 4th Edition. Pearson. 2019. ISBN 013478796X • 9780134787961.

**COURSE TOPICS**

List topics and sequence covered.

Week	Topic
Week 1	Introduction to Java, OOP, and Program Design
Week 2	Primitive data types and their operations
Week 3	Decision and loop control structures, File input and output
Week 4	Methods
Week 5	<b>Midterm Exam 1</b> / Introduction to Classes and Objects, UML diagrams

Week 6	Introduction to Arrays and the ArrayList Class
Week 7	Sorting and Searching Algorithms, Algorithm Analysis
Week 8	More about Classes and Objects
Week 9	<b>Midterm Exam 2</b> / Test Processing and Wrapper Classes
Week 10	Inheritance and Polymorphism
Week 11	Exceptions
Week 12	Recursion
Week 13	Abstract Data Types
Week 14	<b>Final Exam</b>

### NOTES

1. Students are required to follow all College policies. Policies are available on the website at: [Coquitlam College Policies](#)
2. To find out how this course transfers, visit the BC Transfer Guide at: [bctransferguide.ca](http://bctransferguide.ca)