



## Syllabus

### **SCHOOL OF TECHNOLOGY AND COMPUTING** **IS 365: C++ - Introduction**

5 Credits  
Effective: Winter 2013

*Access to the Internet is required.  
All written assignments must be in Microsoft-Word-compatible formats.  
See the library's APA Style Guide tutorial for a list of resources that can help you use APA style.*

## **FACULTY**

Faculty Name: FACULTY NAME

Contact Information: CONTACT INFORMATION

[INSTRUCTOR MAY INSERT PERSONAL MESSAGE IF DESIRED]

## **COURSE DESCRIPTION**

This course provides an introduction to programming using the Standard C++ programming language. Students learn the basic concepts of assignment, iteration, and looping. The course includes extensive coverage of objects and the concepts of object-oriented programming. In addition, students will learn how to effectively structure a program.

## **COURSE RESOURCES**

Required and recommended resources to complete coursework and assignments are available from the [Course Document Lookup](#).

## **CITYU LEARNING GOALS**

This course supports the following City University learning goals:

- Professional competency and professional identity

## **COURSE OUTCOMES**

In this course, learners:

- Apply object-oriented concepts.
- Program effectively in C++.
- Work in a C++ programming environment.
- Follow quality coding practices to produce documented, legible, maintainable, reusable and efficient code.

## **CORE CONCEPTS, KNOWLEDGE, AND SKILLS**

- Assign the data members of one object to those of another object by default memberwise assignment.
- Call a member function of an object to perform a task.
- Complete basic pointer-based string processing.
- Construct programs modularly from functions.
- Control repetition through counters and sentinals.
- Convert objects from one class to another.
- Create a basic program.
- Create basic C++ output and input.
- Create basic flow charts (UML activity diagrams) for sequence, selection, and iteration.
- Create classes that provide overloaded operators.

- Create objects composed of other objects.
- Declare and initialize arrays and refer to the individual elements of arrays.
- Declare and manipulate multidimensional arrays.
- Define a class and use it to create an object.
- Define base classes and derived classes and the relationships between them.
- Define constructors with default arguments.
- Demonstrate decision making using the if statement and relational operators.
- Demonstrate how break and continue alter the flow of control.
- Describe how destructors perform “termination housekeeping” on an object before it is destroyed.
- Describe inheritance and how it promotes software reuse.
- Describe the close relationships between pointers and arrays.
- Describe the distinction between abstract and concrete classes and how to create abstract classes.
- Detail when constructors and destructors are called and the order in which they are called.
- Determine how the visibility of identifiers is limited to specific regions of programs.
- Develop algorithms through top-down, stepwise refinement.
- Distinguish between class templates and class-template specializations.
- Engineer a class to separate its interface from its implementation.
- Explain how polymorphism makes programming more convenient and systems more extensible.
- Explain preprocessor and using Directives.
- Explain the protected member access specifier.
- Explain the similarities and differences between pointers and references and when to use each.
- Form complex conditional expressions.
- Identify the steps of creating a program.
- Implement a class’s attributes as data members.
- Implement a class’s behaviors as member functions.
- Implement basic searching and sorting techniques.
- Implement mechanisms for passing data to functions and returning results.
- Implement multiple selection using the switch statement.
- Install and learn basic functions of the Visual C++ Express.
- Outline how C++ implements virtual functions and dynamic binding.
- Outline the differences between public, protected, and private inheritance.
- Overload function templates.
- Overload unary and binary operators.
- Pass arrays to functions.
- Perform basic arithmetic on variables.
- Review basic terms and concepts in Object Technology.
- Set and use Breakpoints, watch values, and control execution.
- Specify const objects and const member functions.
- Understand basic problem-solving techniques.
- Understand class scope and accessing class members via the name of an object, a reference to an object or a pointer to an object.
- Understand how the function call/return mechanism is supported by the function call stack and activation records.
- Understand the concept of a container class.
- Understand the difference between data members and local variables.
- Understand the role of High-Level Languages in a computer.
- Understand variable concepts of name, type, size and value.
- Understand what pointers are.

- Use a constructor to initialize an object's data.
- Use a preprocessor wrapper to prevent multiple definition errors.
- Use arrays of pointers.
- Use arrays to store, sort and search lists and tables of values.
- Use class templates to create groups of related types.
- Use common math library functions.
- Use constructor and destructors in inheritance hierarchies.
- Use for and do..while to execute statements in a program repeatedly.
- Use friend functions and friend classes.
- Use function templates to create a group of related functions.
- Use header files to manage complexity.
- Use increment, decrement and assignment operators.
- Use inheritance to customize existing software.
- Use objects of class string.
- Use operator overloading to enhance classes.
- Use overload operators and additional features of the C++ string class.
- Use pointers to functions.
- Use proxy classes to hide implementation details from a class's clients.
- Use random number generation.
- Use runtime type information.
- Use static data members and member functions.
- Use the array data structure to represent a set of related data items.
- Use the C++ Standard library class template vector.
- Use the if and if...else selection statements to choose among alternative actions.
- Use the keyword explicit to indicate that a constructor cannot be used for implicit conversions.
- Use the This pointer.
- Use the Visual C++ Express debugger.
- Use the while repetition statement to execute statements in a program repeatedly.
- Write and use recursive functions.

## OVERVIEW OF COURSE GRADING

The grades earned for the course will be derived using City University of Seattle's decimal grading system, based on the following:

<i>Overview of Required Assignments</i>	<i>% of Final Grade</i>
Install Visual C++ and Program	10%
Programming Assignment 2	10%
Programming Assignment 3	10%
Programming Assignment 4	10%
Programming Assignment 5	5%
Programming Assignment 6	5%
Programming Assignment 7	10%
Programming Assignment 8	10%
Programming Assignment 9	5%
Programming Assignment 10	5%
Discussions and Instructor Assignments	20%
<b>TOTAL</b>	<b>100%</b>

## SPECIFICS OF COURSE ASSIGNMENTS

The instructor will provide grading rubrics that will provide more detail as to how this assignment will be graded.

### Install Visual C++ and Program

Install Visual C++ Express. Create basic programs for Exercise 2.15, 2.20, 2.30.

<i>Components</i>	<i>% of Grade</i>
Program C++	30%
Requirements	20%
Programming Environment	50%
<b>TOTAL</b>	<b>100%</b>

### Programming Assignment 2

Exercises 3.13, 3.16.

<i>Components</i>	<i>% of Grade</i>
Object-oriented Concepts	50%
Programming Environment	20%
Program C++	20%
Requirements	10%
<b>TOTAL</b>	<b>100%</b>

### Programming Assignment 3

Exercises 4.13, 4.34, 5.14, 5.29

<i>Components</i>	<i>% of Grade</i>
Program C++	50%
Programming Environment	20%
Quality coding practices	30%
<b>TOTAL</b>	<b>100%</b>

### Programming Assignment 4

Exercises 6.12, 6.36, 6.40

<i>Components</i>	<i>% of Grade</i>
Program C++	50%
Object-oriented Concepts	20%
Quality coding practices	30%
<b>TOTAL</b>	<b>100%</b>

### Programming Assignment 5

Exercises 7.11, 7.17, 7.39.

<i>Components</i>	<i>% of Grade</i>
Program C++	50%
Object-oriented Concepts	30%
Quality coding practices	20%
<b>TOTAL</b>	<b>100%</b>

### Programming Assignment 6

Exercise 8.12

<i>Components</i>	<i>% of Grade</i>
Program C++	70%
Quality coding practices	30%
<b>TOTAL</b>	<b>100%</b>

### Programming Assignment 7

Exercises 9.11, 9.12, 10.10

<i>Components</i>	<i>% of Grade</i>
Object-oriented Concepts	50%
Program C++	30%
Programming Environment	20%
<b>TOTAL</b>	<b>100%</b>

### Programming Assignment 8

Exercises 11.8, 11.10.

<i>Components</i>	<i>% of Grade</i>
Object-oriented Concepts	50%
Program C++	30%
Quality coding practices	20%
<b>TOTAL</b>	<b>100%</b>

### Programming Assignment 9

Exercises 12.7, 12.8

<i>Components</i>	<i>% of Grade</i>
Object-oriented Concepts	50%
Program C++	30%
Quality coding practices	20%
<b>TOTAL</b>	<b>100%</b>

### Programming Assignment 10

Exercises 13.13, 14.3

<i>Components</i>	<i>% of Grade</i>
Object-oriented Concepts	50%
Program C++	30%
Quality coding practices	20%
<b>TOTAL</b>	<b>100%</b>

### Discussions and Instructor Assignments

The instructor will lead course discussions and provide additional assignments as appropriate. Rubrics may be revised appropriately by the instructor.

<i>Components</i>	<i>% of Grade</i>
Quality of responses	30%
Timeliness of responses	20%
Object-oriented Concepts	25%
Program C++	25%
<b>TOTAL</b>	<b>100%</b>

## **COURSE POLICIES**

## **Late Assignments**

LATE ASSIGNMENT

## **Participation**

PARTICIPATION

## **Professional Writing**

Assignments require error-free writing that uses standard English conventions and logical flow of organization to address topics clearly, completely, and concisely. CityU requires the use of APA style.

## **UNIVERSITY POLICIES**

You are responsible for understanding and adhering to all of City University of Seattle's academic policies. The most current versions of these policies can be found in the [University Catalog](#) that is linked from the CityU Web site.

### **Scholastic Honesty**

Scholastic honesty in students requires the pursuit of scholarly activity that is free from fraud, deception and unauthorized collaboration with other individuals. You are responsible for understanding CityU's policy on scholastic honesty and adhering to its standards in meeting all course requirements. A complete copy of this policy can be found in the [University Catalog](#) in the section titled *Scholastic Honesty* under *Student Rights & Responsibilities*.

### **Attendance**

Students taking courses in any format at the University are expected to be diligent in their studies and to attend class regularly.

Regular class attendance is important in achieving learning outcomes in the course and may be a valid consideration in determining the final grade. For classes where a physical presence is required, a student has attended if s/he is present at any time during the class session. For online classes, a student has attended if s/he has posted or submitted an assignment. A complete copy of this policy can be found in the [University Catalog](#) in the section titled *Attendance Policy for Mixed Mode, Online and Correspondence Courses*.

## **SUPPORT SERVICES**

### **Disability Resources**

If you are a student with a disability and you require an accommodation, please contact the Disability Resource Office as soon as possible. For additional information, please see the section in the [University Catalog](#) titled *Students with Special Needs* under *Student Rights & Responsibilities*.

## **Library Services**

CityU librarians are available to help you find the resources and information you need to succeed in this course. Contact a CityU librarian through the [Ask a Librarian](#) service, or access [library resources and services online](#), 24 hours a day, seven days a week.

## **Smarthinking**

As a CityU student, you have access to 10 free hours of online tutoring offered through Smarthinking, including writing support, from certified tutors 24 hours a day, seven days a week. Contact CityU's Student Support Center at [help@cityu.edu](mailto:help@cityu.edu) to request your user name and password.