



Course Programming: Object Oriented Game Programming (2014-2015)

Code / Version PROG2370 (100)

Total Hours 75

Credits 5

PreRequisite(s) PROG1780 (100) Programming: Fundamentals
or PROG1783 (100) IT Support Prog Fundamentals

CoRequisite(s)

Course Description

This course teaches object oriented C# programming through the creation of computer games. Students learn how to create games using the latest development technology. Standards, testing, documentation and practical object-oriented programming techniques are emphasized.

PLAR Eligible: Yes

Course Outcomes

Successful completion of this course will enable the student to:

1. Explain object-oriented programming terms and concepts.
 2. Plan, document, debug, and test object-oriented programs.
 3. Write maintainable code that conforms to coding standards.
 4. Create C# programs using Microsoft Visual Studio.
 5. Create Java programs using NetBeans or Eclipse or similar Integrated Development Environment (IDE)
 6. Write computer games using the latest object oriented game development frameworks.
-

Unit Outcomes

Successful completion of the following units will enable the student to:

- 1.0 Getting Started with C#
 - 1.1 Describe how computers store numbers and characters.
 - 1.2 Explain the relationship between object-oriented analysis, design, programming, and program planning.
 - 1.3 Describe the history and positioning of C#.
 - 1.4 Create, compile, link, and run C# applications using Visual Studio.
 - 1.5 Distinguish between documentation and implementation comments.
 - 1.6 Prepare documentation and implementation comments.
 - 2.0 C# Fundamentals
 - 2.1 Explain classes, objects and instantiation.
 - 2.2 Create syntactically valid, meaningful, and self-documenting C# identifiers.
 - 2.3 Describe C# data types and operators.
 - 2.4 Categorize C# data types and operators.
 - 2.5 Explain the significance of named constants for creating maintainable programs.
 - 2.6 Create C# programs that use the fundamental features of C#.
 - 3.0 C# Methods and Constructors
 - 3.1 Distinguish between call by value and call by reference.
-



Course Programming: Object Oriented Game Programming (2014-2015)

Code / Version PROG2370 (100)

-
- 3.2 Distinguish between classes that do and do not have a Main() method.
 - 3.3 Distinguish between class and instance methods.
 - 3.4 Explain access control modifiers.
 - 3.5 Explain constructors, fields, and properties.
 - 3.6 Identify the signature of a method and explain method overloading.
 - 3.7 Describe equality, relational, and logical operators.
 - 3.8 Create C# programs that define methods and constructors.
 - 4.0 C# Flow of Control and Exception Handling
 - 4.1 Create C# programs that use selection constructs (if...then...else and switch).
 - 4.2 Differentiate between control loops (while), counted loops (for), and foreach.
 - 4.3 Create C# programs that use looping constructs (for, foreach, while, and do...while).
 - 4.4 Create C# programs that demonstrate mastery of structured exception handling.
 - 5.0 Arrays and Collections
 - 5.1 Distinguish between arrays, ArrayList, and generic List.
 - 5.2 Write C# programs that use arrays and collections.
 - 6.0 C# Object-Oriented Programming
 - 6.1 Identify OOP terms, including class, interface, instance, instantiation, encapsulation, polymorphism, inheritance, base/super/parent class and derived/sub/child class.
 - 6.2 Explain the role of access modifiers in inheritance.
 - 6.3 Explain method overriding.
 - 6.4 Explain the significance and features of the Object class.
 - 6.5 Explain interfaces and how they remove the need for multiple inheritances.
 - 6.6 Distinguish between concrete classes, abstract classes, and interfaces.
 - 6.7 Create programs that use the object-oriented feature of C#.
 - 7.0 Getting started with Java
 - 7.1 Discuss Java data types and operators.
 - 7.2 Write, compile and run Java console applications by using a Java Integrated Development Environment (IDE), and at the command line.
 - 7.3 Convert input string values to numerical data and vice versa.
 - 7.4 Write string processing programs using String and StringBuilder objects
 - 7.5 Write javadoc comments for classes and methods, and generate documentation.
 - 7.6 Write code that implements basic search and sort algorithms by using loops and array
 - 7.7 Explain threads in Java.
 - 8.0 Java Object Oriented Programming
 - 8.1 Define a class with multiple methods and data members.
 - 8.2 Discuss how objects are compared
 - 8.3 Organize classes into packages
 - 8.4 Create Java GUI application that uses user-defined event handlers.
 - 8.5 Manipulate a collection of objects using array lists, linked lists and maps.
 - 8.6 Describe the basic operations of a queue and stack.
 - 8.7 Create programs where polymorphism is applied by interfaces, abstract methods and method overriding.
-



Course Programming: Object Oriented Game Programming (2014-2015)

Code / Version PROG2370 (100)

9.0 Exception handling and testing Java programs

- 9.1 Write programs with exception handling
- 9.2 Create programmer defined exceptions
- 9.3 Debug Java program using IDE.
- 9.4 Write test methods using JUnit framework, and run test suite using Java IDE

10.0 Game Programming

- 10.1 Create game programs that display text.
- 10.2 Create game programs that display and animate 2D and 3D objects.
- 10.3 Create game programs that respond to keyboard and mouse events.
- 10.4 Create game programs that persists state, configuration and history.
- 10.5 Create game programs that implement basic game artificial intelligence and game physics.
- 10.6 Create games with sound effects and music.

Required Student Resources

Optional Student Resources

Rob Miles. Microsoft XNA Game Studio 4.0: Learn Programming Now!. Microsoft Press.

The book is available on Safari. Go to <http://www.conestogac.on.ca/lrc/resources/>, click on Safari, and search by ISBN

Adam Dawes. Windows 8 and Windows Phone 8 Game Development. Apress.

The book is available on Safari. Go to <http://www.conestogac.on.ca/lrc/resources/>, click on Safari, and search by ISBN

Joel Murach and Andrea Steelman. Murach's Java SE 6 (April 2007).

Evaluation

The minimum passing grade for this course is 55 (D).

In order to successfully complete this course, the student is required to meet the following evaluation criteria:

Assignments	40.00
Midterm Tests	30.00
Final Exam, Projects	30.00
	<hr/>
	100.00 %

Other

Conestoga College is committed to providing academic accommodations for students with documented disabilities. Please contact the Accessibility Services Office.

Prepared By Sabbir Ahmed

School Information Technology

Date 2014-08-12

© Conestoga ITAL



Course Programming: Object Oriented Game Programming (2014-2015)

Code / Version PROG2370 (100)
