
Course Outline

for

Object-Oriented Programming

Version 1.0

Prepared by: Fawad Ishaq Ch

Organization: NTI

Date created: May 29, 2008

Table of Contents

Table of Contents	ii
1. Introduction	1
1.1 Abstract	1
1.2 About the Instructor	1
1.3 Introduction	1
1.4 Learning Objectives	1
1.5 Web References	2
2. Course Plan	2
2.1 Course Contents	2
2.2 Teaching Methodology	3
2.3 Recommended Text	3
2.4 Course Material	3
2.5 Evaluation Criteria	3
2.6 Development Environment	3

1. Introduction

1.1 Abstract

The primary objective of this document is to specify the course outline for the **Object-Oriented Programming** course.

1.2 About the Instructor

Fawad Ishaq Ch is currently working as a Business Analyst at Innovative (PVT) Ltd. one of the leading card processors in the world. He obtained his MCS degree from the Hamdard University, Karachi. His teaching experience includes 7 years as lecturer computer science and information technology at various leading universities in the country. He also has an industry experience of 3 years as system analyst, one of which was spend as Manager IT.

1.3 Introduction

Object-oriented programming (OOP) is a currently popular and powerful programming technique. The main characteristics of OOP are encapsulation, inheritance, and polymorphism. C++ accommodates OOP by providing classes, a kind of data type combining both data and algorithms. C++ is not what some authorities would call a “pure OOP language.” C++ tempers its OOP features with concerns for efficiency and what some might call “practicality.” This combination has made C++ currently the most widely used OOP language, although not all of its usage strictly follows the OOP philosophy.

1.4 Learning Objectives

- Enable students solve real-life problems.
- To develop object-oriented program design (OOD) skills.
- To develop object-oriented program analysis (OOA) skills.
- How to define a class and use it to create an object.
- How to engineer a class to separate its interface from its implementation and encourage reuse.
- How inheritance promotes software reuse.
- To create classes by inheriting from existing classes.

- *What polymorphism is? How it makes programming more convenient, and how it makes systems more extensible and maintainable.*

1.5 Web References

- <http://fawadishaq.wordpress.com>

2. Course Plan

2.1 Course Contents

- *Object Oriented Programming Concepts*
Topics: Object-oriented paradigm, data abstraction, encapsulation, inheritance, Polymorphism.
Activities: Classroom exercise, Home Assignment.
- *Introduction to Classes and Objects*
Topics: classes, objects, data members, member functions
Activities: Interactive learning, Lab exercises
- *Classes Advanced*
Topics: friends, static, composition, this, const
Activities: Interactive learning, Lab exercises
- *Operator overloading*
Topics: stream insertion, stream extraction, binary operator, unary operator
Activities: Interactive learning, Lab exercises
- *Inheritance*
Topics: single inheritance, multiple inheritances, protected members, method over-riding
Activities: Interactive learning, Lab exercises
- *Polymorphism*
Topics: virtual function, pure-virtual functions, abstract class, abstract super class
Activities: Interactive learning, Lab exercises
- *Standard Template Library (STL)*
Topics: stack, queue, list, vector
Activities: Interactive learning, Lab exercises
- *File Processing*
Topics: files & streams, sequential access files, random access files
Activities: Interactive learning, Lab exercises, Project

2.2 Teaching Methodology

- Classroom-based teaching
- Classroom-based training
- Group activities & discussions
- Q & A sessions

2.3 Recommended Text

- C++ how to program, Deitel & Deitel, 5th edition
- Thinking in C++, Bruce Eckel, 2nd edition

2.4 Course Material

- Instructor handouts
- Lab exercises
- Internet exercises
- Web references

2.5 Evaluation Criteria

Evaluation is strictly based on how well the student will perform in class activities, Lab activities, and other activities. Performance will be monitored throughout the semester. Following are the tools used to measure the competency level of the candidate:

- Class quizzes (MCQ type). Normally conducted at the end of each unit.
- Lab exercises
- Home Assignments
- Projects

2.6 Development Environment

- Windows XP
- Visual C++ 6.0