COMSATS-Lancaster Dual Degree Programme

Lancaster Block, CIIT, Defence Road, Off Raiwind Road, Lahore

LANCASTER UNIVERSITY

COURSE HANDBOOK

©www.fall11.co.cc

1	**Course Title**	Object Oriented Programming
2	**Course Code**	CSC245
3	**Credit Hours**	4(3,1)
4	**Semester**	Spring 2012
5	**Resource Person**	Dr.Tabbasum Naz
6	**Supporting Team Members**	Ayesha Sadiq
7	**Contact Hours (Theory)**	3 hours per week
8	**Contact Hours (Lab)**	3 hours per week
9	**Office Hours**	10:00 AM to 11:30 PM, Mon, Thru, Fri

10 Course Introduction

Object-oriented programming (OOP) is a programming paradigm that uses "objects" – data structures consisting of data members and methods together with their interactions – to design applications and computer programs using Java development environment. Programming features includes object oriented design strategy and problem solving, objects and classes, member functions, public and private members, constructors and destructors, data abstraction, data encapsulation, data modularity, method overloading, method overriding, polymorphism, inheritance, interfaces, exception Handling and Garbage Collection. It includes understanding of graphical user interface (GUI). Create and manipulate basic components like labels, lists, panels, buttons used in building GUI. The students would implement all these concepts in the lab keeping in mind the real-world problems.

11 Learning Objectives

12 Course Contents

The lab part will use more of a supervisory model in which students will be assigned problems related to the theory part that they had covered in class. They will be encouraged to come up with their own solutions and implement them. If and when they get stuck the instructor will guide them through the debugging process. During Lab work students will be able to implement object oriented programming concepts using real time case studies and examples; it includes solution of the problems faced by student during implementation of taught concepts and tries to implement problems using good programming practices and processes, So that they can develop efficient and robust software in a cost-effective manner in the future.

13 Lab Weekly Plan

Weeks	**Topic of Lab**	**Reading Assignment**
Week 1	Difference between JAVA and C++	Beginning Java 2 By Ivor

	Types of java programs Applets Java Applications • Console Based Applications • Windowed Application The Java Execution Environment The Java Development Environment • Installing JDK • Running Java Application First Java Display Program (Hello Word) using java basic program structure).	Horton Chapter1+ Handouts
Week 2 & 3	Demonstration of use of primitive java data types in java. Use of Access Specifiers with data types in java. Develop a Program for performing arithmetic Operations on numbers. Developing a Class with Attributes and methods Demonstration of modelling real word objects in classes and their usage with different control and iterative statements.	Chapter 2
Week 4	Use of getter setter methods Use of instance variable and class variable (Declaration and access). Use of instance methods and class methods (Define and access). Use of default Constructor. Write a program to calculate area of the following shapes using classes and object concept for each shape. • Box • Circle • Rectangle • Square	Chapter 3
Week 5	A short Introduction to Java Class Libraries. Use of Parameterized Constructor	Chapter 6 + Chapter 8

	Use of Multiple Constructors in the same class. Write a program to calculate volume, perimeter, area of the following shapes using classes and multiple constructors concepts. • Cube • Rectangle • Triangle	
Week 6	Use of Objects as a parameter passing. Use of Arrays and Random Number generation. Write a program for the class Hat having following data members of following types and getter and setter methods of each member. Note: Specify three objects of type Hat Data members: String ownerName; String color; int size; boolean hatOn Write a program to generate random numbers from 1-100 using initialization blocks for array initialization and print their squares.	Beginning Java 2 By Ivor Horton, Chapter 7
Week 7	Method overloading Example Overloaded Constructor Write a program to test method overloading using different parameter passing	Chapter 6
Week 8 & 9	Demonstration for Inheritance Concept A simple example of inheritance using super class and subclass. • Box as Super class • BoxColor as base class Inheritance Concept using default, parameterized and Constructor A simple example of inheritance using super class A and Child class B to access data members and functions of parent class by child class. A example of inheritance defining how parent class constructor is being called within the child class's constructor using default, parameterized and Constructor	Chapter 9

	that get an object of the parent class as a parameter.	
Week 10	Method Overriding Contents An implementation to show that Methods with differing type signatures are overloaded – not Overridden. Applying method overriding Dynamic Method Dispatching	Chapter 10
Week 11	Run time Polymorphism Implementation to show that if a super class contains a method that is overridden by a subclass, then when different types of objects are referred to through a super class reference variable, different version of the method are executed. A program creates a super class called Figure that stores the dimensions of various two-dimensional objects. It also defines a method called area ( ) that computes the area of an object. The program derives two subclasses from Figure. The first is Rectangle and the second is Triangle. Each of these subclasses overrides area( ) so that it returns the area of a rectangle and a triangle, respectively	Chapter 11 & 12
Week 12	Use of Interfaces Demonstration for Multiple Inheritance by interfaces	
Week 13	Examples to implement Exception Handling in different scenarios.	Chapter 13
Week 14	Examples for String Handling in Java	The Complete Reference JAVA2 Chapter 13

Week 15 & 16	File Handling Examples • Reading from file • Writing to files • Copy two files • Searching in files Accessing Database with JDBC examples for inserting, deleting, searching from a data base.	The Complete Reference JAVA2 Chapter 12 + Chapter 25 Text Book

14 Course Assessment

The assessment of this module shall have following breakdown structure

Lab Part

Sessional-I Exam	10%
Sessional-II Exam	15%
Final Exam	50%
Quiz (3-6 per semester)	10%
Assignments\Projects (3-6 per semester)	10%
Lab Tasks	5 %

The minimum pass marks for each course shall be 50%. Students obtaining less than 50% marks in any course shall be deemed to have failed in that course. The correspondence between letter grades, credit points, and percentage marks at CIIT shall be as follows:

Grades	Letter Grade	Credit Points	Percentage Marks
A	(Excellent)	4.0	90and above
A-		3.7	85-89
B+		3.3	80-84
B	(Good)	3.0	75-79
B-		2.7	70-74
C+		2.3	65-69
C	(Average)	2.0	60-64
C-		1.7	55-59
D	(Minimum passing)	1.3	50-54
F	(Failing)	0.0	Less than 50

Note: The marks to be assigned to students shall be in whole numbers and are not same as followed in the annual system of Lancaster University.

16 Assessment Schedule

Week	**Assignment and Quiz**
Week 2	1st Assignment and Quiz
Week 5	2nd Assignment and Quiz
Week 8	3rd Assignment and Quiz
Week 11	4th Assignment and Quiz
Week 13	5th Assignment (Project) and Quiz
Week 15	6th Assignment (Presentation) and Quiz

17	Format of Assignment

This course indoctrinates the following format for all assignments except code:

1. Paper Size: A4
2. Left Margin: 2 Inches
3. Right Margin: 1 Inch
4. Top Margin: 0.5 Inch
5. Bottom Margin: 0.5 Inch
6. Font: Times New Roman
7. Font Size:
    a. Main Heading 14
    b. Sub Heading 12
    c. Text 12
    d. Titles 16
8. Font Color: Black
9. Line Spacing: 1.5
10. Diagrams & Charts: Need not be colored
11. Title page must be designed as guided by resource person in class
12. Number of Pages: No Limit
13. Reference Style: APA (If applicable)

Code will be submitted in text files that can be compiled with specified programming language compiler.

PROGRAMMING EXERCISES

Several programming exercises will be given throughout the course. The purpose of these exercises will be to give students practical knowledge of the concepts taught in the class. This hands-on practice should give students better understanding of the course material and enhance their software development skills.

The submittal for every programming exercise should include the following items.

- Hard and Soft copies of the program source code
- Results of running the program
- Input data used for program run(s)
- Signed disclaimer that the work is your own (If it is determined that two or more students have collaborated, then those students may fail the course regardless of who copied from whom)
- All documents must be bound together (unbound submittals will not be accepted)
- For some exercises, a written documentation will also be required. The details of this report's contents will be given in the handout for the programming exercise.

18	**Text Book**	• "Java How to Program", 7th Edition by Deitel & Deitel, ISBN: 0132222205
19	**Reference Books**	• "Java 2: The Complete Reference", by Patrick Naughton and Herbert Schildt • "Thinking in Java" by Bruce Eckel, Prentice Hall, 4th Edition, 2006 • "Beginning Java 2" by Ivor Horton • "Object-Oriented Software Development Using Java", --2nd Editions, Addison Wesley 2003

		- “The Elements of Java Style”, By Allan Vermeulen, et al.

20 Plagiarism

Plagiarism involves the unacknowledged use of someone else’s work, usually in coursework, and passing it off as if it were one’s own. Many students who submit apparently plagiarised work probably do so inadvertently without realising it because of poorly developed study skills, including note taking, referencing and citations; this is poor academic practice rather than malpractice. Some students, particularly those from different cultures and educational systems, find UK academic referencing/acknowledgement systems and conventions awkward, and proof-reading is not always easy for dyslexic students and some visually-impaired students. Study skills education within programmes of study should minimise the number of students submitting poorly referenced work. However, some students plagiarise deliberately, with the intent to deceive. This intentional malpractice is a conscious, pre-mediated form of cheating and is regarded as a particularly serious breach of the core values of academic integrity. **The Dual Degree Prorgamme has zero tolerance for intentional plagiarism.**

Plagiarism can include the following:

1. Collusion, where a piece of work prepared by a group is represented as if it were the student’s own;
2. Commission or use of work by the student which is not his/her own and representing it as if it were, e.g.:
    a. purchase of a paper from a commercial service, including internet sites, whether pre-written or specially prepared for the student concerned
    b. submission of a paper written by another person, either by a fellow student or a person who is not a member of the university;
3. Duplication (of one’s own work) of the same or almost identical work for more than one module;
4. The act of copying or paraphrasing a paper from a source text, whether in manuscript, printed or electronic form, without appropriate acknowledgement (this includes quoting directly from another source with a reference but without quotation marks);
5. Submission of another student’s work, whether with or without that student’s knowledge or consent;
6. Directly quoting from model solutions/answers made available in previous years;
7. Cheating in class tests, e.g.
    a. when a candidate communicates, or attempts to communicate, with a fellow candidate or individual who is neither an invigilator or member of staff
    b. copies, or attempts to copy from a fellow candidate
    c. attempts to introduce or consult during the examination any unauthorised printed or written material, or electronic calculating, information storage device, mobile phones or other communication device
    d. personates or allows himself or herself to be impersonated.
8. Fabrication of results occurs when a student claims to have carried out tests, experiments or observations that have not taken place or presents results not supported by the evidence with the object of obtaining an unfair advantage.

These definitions apply to work in whatever format it is presented, including written work, online submissions, group work and oral presentations.

21 Attendance Policy

Every student must attend 80% of the lectures/seminars delivered in this course.

22	**Field Trips/Case Studies/Seminars/Workshop**
Not Applicable	