

Title (Units): **COMP1150 Object Oriented Programming (3,3,2)**

Course Aims: To study the object-oriented programming principles and techniques. Upon completion, students should be able to use an object-oriented language to develop rather complex programs.

Prerequisite: COMP 1180 Structured Programming or COMP1170 Introduction to Structured Programming

Learning Outcomes (LOs):

Upon successful completion of this course, students should be able to:

No.	Learning Outcomes (LOs)
	Knowledge
1	Describe the principles of object-oriented programming
2	Apply the concepts of data encapsulation, inheritance, and polymorphism to large-scale software
3	Acquire the concepts of Graphical User Interfaces
	Professional Skill
4	Design and develop object-oriented computer programs
5	Design and develop programs with Graphical User Interfaces capabilities
	Transferable Skill
6	Formulate problems as steps so as to be solved systematically
	Attitude
7	Integrate robustness, reusability, and portability into large-scale software development
8	Develop software with team-work in mind

Calendar Description: This course introduces the object-oriented programming concepts, principles, and techniques, including classes, objects, inheritance, and polymorphism. All these concepts are illustrated via a contemporary object-oriented programming language.

Assessment:

No.	Assessment Methods	Weighting	Remarks
1	Continuous Assessment	40%	Continuous assessments are designed to measure how well the students have learned the fundamentals and major concepts of object-oriented programming. A number of machine problems will be given to students to train them to design programs via the object-oriented approach. Practical tests will be used to test their programming capabilities.
2	Examination	60%	Final examination questions are designed to see how far students have achieved their intended learning outcomes. Questions will primarily be concepts and skills based to assess the student's ability in object-oriented programming.

Rubrics

Criteria	Excellent (A)	Good (B)	Satisfactory (C)	Marginal Pass (D)	Fail (F)
Principles of object-oriented programming	The student acquires excellent knowledge in the principles of object-oriented languages, namely, data encapsulation, inheritance, and polymorphism.	The student acquires sufficient knowledge in the principles of object-oriented languages, namely, data encapsulation, inheritance, and polymorphism.	The student acquires average knowledge in the principles of object-oriented languages, namely, data encapsulation, inheritance, and polymorphism.	The student is able to describe the meanings of data encapsulation, inheritance, and polymorphism, and to give simple examples on them.	The student is unable to describe the meanings of data encapsulation, inheritance, and polymorphism, and to give simple examples on them.
Applying object-	The student is	The student is	The student is	The student can	The student cannot

oriented techniques to software packages	able to extensively apply object-oriented techniques to write software applications with multiple classes, e.g., enforcing data hiding as much as possible via class privacy.	able to sufficiently apply object-oriented techniques to write software applications with multiple classes, e.g., enforcing data hiding via class privacy.	able to apply object-oriented techniques in some key elements of software applications with multiple classes, e.g., enforcing data hiding via class privacy.	apply some object-oriented techniques to write software applications with multiple classes, e.g., enforcing data hiding via class privacy.	apply object-oriented techniques to write software applications with multiple classes, e.g., enforcing data hiding via class privacy.
Graphical user interfaces	The student demonstrates excellent know-how in writing programs with graphical user interfaces.	The student demonstrates considerable know-how in writing programs with graphical user interfaces.	The student demonstrates average know-how in writing programs with graphical user interfaces.	The student demonstrates some know-how in writing programs with graphical user interfaces.	The student does not demonstrate any know-how in writing programs with graphical user interfaces.
Exception handling	The student correctly writes object-oriented programs with complicated exception handling facilities.	The student correctly writes object-oriented programs with considerable exception handling facilities.	The student correctly writes object-oriented programs with an average amount of exception handling facilities.	The student correctly writes object-oriented programs with some exception handling facilities.	The student cannot write object-oriented programs with any exception handling facilities.

Learning Outcomes and Weighting:

Content	LO No.
I. Object-oriented Concepts	1-2, 4-5, 7-8
II. Object-oriented Programming	1-2, 4-5, 7-8
III. GUI Programming	3, 6
IV. Exception Handling, Streams and Files, and Advanced Features	7-8

- References:**
- C. S. Horstmann and G. Cornell, Core Java 2 (Volume I-Fundamentals), Prentice Hall, 7th Edition, 2004.
- H. M. Deitel and P. J. Deitel, Java How to Program, Prentice Hall, 7th Edition, 2007.
- A. Kak, Programming with Objects: A Comparative Presentation of Object Oriented Programming with C++ and Java, Wiley-IEEE Press, 2003.
- J. Bishop, Java Gently: Programming Principles Explained, Addison-Wesley, 3rd Edition, 2001.
- G. Booch, R. A. Maksimchuk, M. W. Engel, and B J. Young, Object-oriented Analysis and Design with Applications, Addison-Wesley, 3rd Edition, 2007.
- K. Arnold, J. Gosling, and D. Holmes, Java Programming Language, Prentice Hall, 4th Edition, 2005.

Course Content in Outline:

- Topic**
- I. Object-oriented Concepts
 - II. Object-Oriented Programming
 - A. Classes and objects
 - B. Methods and messages
 - C. Classification, generalization and specialization
 - D. Inheritance
 - E. Interfaces and inner classes

F. Polymorphism

G. modularity

III. GUI Programming

A. Graphical user interface

B. Event handling

C. Applets

IV. Others

A. Exception handling

B. Streams and files

C. Advanced features