## Course Outline

Course code: SWE233

Course Title: Object Oriented Concepts & Design

Course Teacher: Farzana Sadia

Course Teacher Initial: FS

# Course Description:

Object Oriented Concepts & Design includes the concepts Object Oriented Programming (OOP) as a Paradigm or Model of programming and Object Oriented Design Principles as well (not the Object Oriented Design Pattern). It allows of code organization in general and Object Oriented design in particular. Students are intended to break down problems from structured & semi-structured requirements such as a class-diagram or scenario (not a user-story).

# Course Objectives (CO’s)

1. The students should learn the philosophy of Object Oriented Programming (OOP) as a Concept or Model of programming. Starting with the basic concept of Class & Object based on real life example.
2. The students would learn the basic features/ principles of OOP namely Encapsulation, Inheritance, Polymorphism & Abstraction and how to implement these in program (code).
3. The students should learn to combine all the features of OOP. They would relate all these as part of a class project and should complete the course with a small project.

# Prerequisite

SWE132 Introduction to Java Programming

# Learning Outcomes (LO’s)

**By the end of this course students would be able to -**

|  |  |
| --- | --- |
| No. | Outcomes (LO’s) |
| **Knowledge** | |
| 1 | Describe the importance of code organization in general and Object Oriented design in particular |
| 2 | Explain the basic features/ principles of OOP namely Encapsulation, Inheritance, Polymorphism & Abstraction, including why & how to use them |
| 3 | Relate all the features together to combine the concepts altogether |
| **Skill** | |
| 4 | Break down problems from structured & semi-structured requirements |
| 5 | Design a program structure or class diagram from a given scenario |
| 6 | Solve the problems using all the features of OOP concepts |
| **Self Development** | |
| 7 | Demonstrate learned concepts by developing a simple project in OOP |
| **Role in Context** | |
|  |  |
| **Attitude** | |
| 8 | Develop confidence for braking down and solving problems |

# Rubrics

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Weighing | Letter Grade | Category | Description |
| 1 | 80% | A+ | Outstanding | Very Strong evidence of having achieved all the LO’s and demonstration of exceptional mastery of Object Oriented Programming (OOP) knowledge and skills.  Able to develop correct programs to solve problems.  Demonstration of exceptional mastery of OOP concepts & design. |
| 2 | 75% | A | Excellent | Strong evidence of having achieved all the LO’s and demonstration of mastery of OOP knowledge and skills.  Able to develop correct programs to solve problems.  Demonstration of mastery of OOP concepts & design. |
| 3 | 70% | A- | Very Good | Evidence of having achieved 90% of the LO’s with good understanding of OOP knowledge and skills.  Able to develop correct programs to solve problems.  Demonstrate a complete level of OOP concepts & design. |
| 4 | 65% | B+ | Good | Evidence of having achieved 80% of the LO’s with understanding of OOP knowledge and skills.  Able to develop correct programs to solve problems.  Demonstrate a complete level of OOP concepts & design. |
| 5 | 60% | B | Satisfactory | Evidence of having achieved 70% of the LO’s with basic understanding of OOP knowledge and skills.  Able to develop acceptable solution to solve problems.  Demonstrate a adequate level of OOP concepts & design. |
| 6 | 55% | B- | Above Average | Evidence of having achieved 60% of the LO’s with minimal understanding of OOPknowledge and skills.  Able to provide solution to simple problems  Demonstrate a basic level of OOP concepts & design |
| 7 | 50% | C+ | Average | Evidence of having achieved 50% of the LO’s with minimal understanding of OOP knowledge and skills.  Able to provide solution to simple problems.  Demonstrate a basic level of OOP concepts & design. |
| 8 | 45% | C | Below Average | Evidence of having achieved 40% of the LO’s with minimal understanding of OOP knowledge and skills.  Able to provide solution to very simple problems.  Demonstrate a low level of OOP concepts & design. |
| 9 | 40% | D | Pass | Evidence of having achieved 30% of the LO’s with little understanding of OOP knowledge and skills.  Able to provide solution to very simple problems.  Demonstrate a very lower level of OOP concepts & design. |
| 10 | <40 | F | Fail | Evidence of having achieved below 30% of the LO’s with very little understanding of OOP knowledge and skills.  Unable to provide solution to very simple problems.  OOP concepts & design skills falling below the basic minimum level. |

# Reference:

1. Object-Oriented Programming and Java Second edition by Danny Poo Derek, Kiong Swarnalatha Ashok
2. The Essence of Object Oriented Programming with Java and UML by Bruce E. Wampler, Ph.D.
3. An Introduction to Object-Oriented Programming 3rd Edition by Timothy A. Budd
4. Java: How to Program, 9th Edition (by Deitel)
5. Java: The Complete Reference, 9th Edition (by H. Schield)

# Assessment Policy:

**Quiz:** Three quizzes will be taken, all will be counted.

1st Quiz: Last week of May

2nd Quiz: Second week of June

3rd Quiz: Last week of July

**Assignment**: Five assignments of hard copy must be submitted. NO LATE SUBMISSION IS ALLOWED.

Assignment-1 submission: Before Mid (Encapsulation, Inheritance)

Assignment-2 submission: After Mid(Polymorphism, Abstraction)

**Lab Performance**: Five lab performances will be taken on instant problem solving.

**Lab Test:** Two lab tests will be taken.

**Project:**  A project must be developed using object oriented concepts and graphical user interfaces having at least five modules, at least ten classes. **Project Submission: First week of August**

**Mark Distribution: Total 100 Marks (Theory: 75 Marks + Lab 25 Marks)**

|  |  |
| --- | --- |
| ***Theory:75 marks*** | ***Lab: 25 Marks*** |
| **Class Attendance: 5** | **Lab performance: 5** |
| **Assignment: 5** | **Lab Test: 5** |
| **Quiz: 15** | **Lab Final Exam: 5** |
| **Mid Exam: 20** | **Project & Viva: 10** |
| **Final Exam: 30** |  |

**Course Outline:**

* + Methods
  + Pass By Reference
  + Encapsulation
  + Inheritance
  + Polymorphism
  + Abstraction

# Attendance policy:

**Attendance marks will be given by DIU ERP.**

# Disclaimer:

While every effort has been made to ensure that the information contained in this document is accurate, the information is subject to change. Changes will be notified in class and/or tutorials, via Google Classroom or email. Students are encouraged to check Google Classroom or email for any changes. It is your responsibility to be informed.