



Davao del Sur State College
Brgy. Matti, Digos City, Davao del Sur

Object-Oriented Programming

Version 2.0





Object-Oriented Programming

Version 2.0

**A Course Pack for
BSIT, BSIS and DIT Students**

Razel Ann G. Aguilar, MIT
Davao del Sur State College
Matti, Digos City, Davao del Sur



COURSE OVERVIEW

Hello dear students! Welcome to your journey into Object-Oriented Programming (OOP), an exciting step forward from the programming concepts you already know. In this course, we'll build on your background in procedural programming and guide you toward a more powerful and organized way of creating programs.

We'll begin by reviewing the basic concepts of conditions and loops, as well as data types, with an emphasis on structured data types, arrays, pointers, and functions. From there, we'll gradually introduce you to the world of object-oriented programming, where you'll learn how real-world problems are modeled using classes, objects, and methods. Along the way, you'll discover core OOP principles like abstraction, encapsulation, inheritance, and polymorphism, which are essential skills for modern software development. You'll also explore exception handling, templates, file handling, and database connectivity to make your programs more reliable and practical.

This course pack is organized into four modules designed to guide you step by step:

- Module 1: Introduction to Object-oriented Programming
- Module 2: Object-Oriented Programming in Java
- Module 3: Object-Oriented Design Principles
- Module 4: File Handling and Database Connectivity

At the end of this course pack, the learner will be able to:

1. Compare and contrast the procedural approach with the object-oriented programming approach.
2. Design, implement, test, and debug programs using OOP concepts such as abstraction, encapsulation, inheritance, and polymorphism.

To get the most out of this learning experience, we encourage you to complete each lesson in sequence, as every topic builds upon the previous one. Take your time, collaborate with your classmates, and enjoy the learning process. Remember, you're not alone in this journey. Feel free to ask your instructors, or classmates for guidance and support whenever you need it.

Good luck, have fun, and let's start coding!

PREFACE

In today's fast-changing world of technology, dear students, learning object-oriented programming (OOP) is more than just picking up new coding skills, it's about developing a new way to think, design, and solve problems. Moving from procedural programming to OOP allows you to view programs as collections of interacting objects, much like how systems work in the real world.

This course is designed to guide you step by step in building a strong foundation in object-oriented concepts, which are at the core of modern software development. This course pack was carefully developed to provide you with a clear and structured learning pathway, using Java as the primary programming language to support your growth and understanding.

Throughout this course pack, learning is progressive and practical. Each module builds upon the previous one through examples, exercises, and hands-on coding activities that reinforce key ideas. You are encouraged to actively engage with the lessons and apply what you learn through practice, as this will strengthen your problem-solving skills and deepen your appreciation for well-designed object-oriented software.

To make the most of this course pack, students are encouraged to:

- Follow the modules sequentially to build a coherent and cumulative understanding of OOP.
- Practice consistently by working on the coding examples and activities provided in each module.
- Engage with peers and instructors to discuss concepts, troubleshoot challenges, and exchange ideas.
- Review and revisit previous modules if concepts feel unclear or need reinforcement before moving on.

This course pack is more than a collection of theories, it is a learning companion for you, dear students, designed to build confidence and competence in object-oriented programming. Learning OOP may come with challenges, but with practice, collaboration, and persistence, you will see how it simplifies complex problems and inspires creative solutions. Move at your own pace as you explore the concepts, practice the code, and ask questions whenever needed. Your journey into object-oriented programming begins here.

Institute of Computing, Engineering and Technology
Computing Department

First Semester, Academic Year 2024-2025

COURSE OUTLINE

Course Name: IT 622 – OBJECT ORIENTED PROGRAMMING

Course Credits: 3.0

Course Description: This course introduces object-oriented programming concepts to students with a background in the procedural paradigm. The course begins with a review of the basic concepts of conditionals and loops, and data types with an emphasis on structured data types, arrays, pointers, and functions. It then evolves on the object-oriented programming paradigm, focusing on the definition and use of classes, objects, and methods together with the fundamentals of object-oriented design. Other areas include fundamentals of OOP concepts, templates, exceptions, and file handling.

Contact Hours/ Lecture (2 hours per week)

Week: Laboratory (3 hours per week)

Pre-requisite: CS 612 – Computing Programming 2

Course Outcomes: 1. CO1. Compare and contrast the procedural approach with the object-oriented programming approach
2. CO2. Design, implement, test, and debug programs using OOP concepts such as abstraction, encapsulation, inheritance, and polymorphism.

COURSE OUTLINE AND TIMEFRAME

Week	Topic and Sub-Topics	Student Learning Outcomes
1	Course Introduction & Institutional Orientation DSSC's Vision, Mission, and Core Values Classroom Policies, Grading System Course Platform & Tools Setup	At the end of the lesson, the students will be able to: - Recall the Vision, Mission, Goals, and Core Values of the institution. - Appreciate the relevance of the institution's VGMO. - Be oriented in the class policies, course requirements, course assessment, and grading system MAP: Institutional Alignment
	Module 1 – Introduction to	At the end of the lesson, the students will be able to: Analyze the key characteristics, advantages, and limitations of

2	Object-Oriented Programming Lesson 1: Procedural vs. Object-oriented Approach Lab Activity: Comparative Analysis of Procedural and OOP using tables	procedural and object-oriented programming. • Demonstrate appreciation for the principles of both approaches by recognizing their appropriate use in real-world scenarios. • Construct simple code examples that demonstrate both procedural and object-oriented approaches. MAP: CO1
3-5	Module 1 – Introduction to Object-Oriented Programming Lesson 2: The building Blocks of Java: A Review of Fundamentals Concepts Lab Activity: • Hands-on Lab Activity (Instructor demo followed by student practice) • Pair programming activity	At the end of the lesson, the students will be able to: • Explain and apply the fundamental concepts of Java including tokens, control structures, and error types in simple code • Show a positive attitude toward writing clean and error-free Java code • Write and debug Java programs using proper syntax, control flow and error handling. MAP: CO1
6	Module 2 – Object-Oriented Programming in Java Lesson 1: Objects and Classes Lab Activity:	At the end of the lesson, the students will be able to: • Explain and apply the fundamental concepts of Java including tokens, control structures, and error types in simple code • Show a positive attitude toward writing clean and error-free Java code

	Hands-on Coding Activities on objects and classes	Write and debug Java programs using proper syntax, control flow and error handling. MAP: CO1
7-8	Module 2 – Object-Oriented Programming in Java Lesson 2: Encapsulation Lab Activity: Hands-on Coding Activities implementing encapsulation in Java	At the end of the lesson, the students will be able to: - Explain the concept of encapsulation and apply access modifiers to control data visibility and access in Java classes. - Recognize the importance of encapsulation in promoting data security - Implement encapsulation in Java by defining attributes as private and providing appropriate getter and setter methods. MAP: C02
9	Midterm Examination Covers Modules 1 and 2 Lecture and Laboratory Exam	Evaluate students' knowledge and skills relevant to all topics discussed in Modules 1 and 2.
10	Module 3 – Object-Oriented Programming Design Principles Lesson 1: Abstraction Lab Activity: Hands-on coding activities implementing abstraction in Java	At the end of the lesson, the students will be able to: - Explain the concept of abstraction and differentiate it from encapsulation, with examples of abstract classes and interfaces in Java. - Recognize the value of abstraction in simplifying complex systems - Design and implement Java programs using abstract classes or interfaces. MAP: CO2

11-12	Module 3 – Object-Oriented Programming Design Principles Lesson 2: Inheritance Lab Activity: • Hands-on coding activities implementing inheritance in Java • Create superclasses and multiple subclasses	At the end of the lesson, the students will be able to: • Explain the concept of inheritance and analyze how subclasses acquire properties and behaviors from superclasses in Java. • Recognize the value of inheritance in reducing redundancy, and express willingness to reuse code through superclass design in programming. • Implement inheritance in Java by creating subclasses that inherit from a superclass. MAP: CO2
13-14	Module 3 – Object-Oriented Programming Design Principles Lesson 3: Polymorphism Lab Activity: • Hands-on coding activities implementing polymorphism in Java	At the end of the lesson, the students will be able to: • Explain the concept of polymorphism and distinguish between method overloading and method overriding in Java. • Analyze how polymorphism enhances flexibility and scalability in software development. • Implement polymorphism in Java programs using method overriding and overloading. MAP: CO2
	Module 4– File Handling and Database Connectivity Lesson 1: File Input/Output Streams	At the end of the lesson, the students will be able to: • Explain the purpose and types of file input/output streams in Java and differentiate between byte and character streams.

15	Lab Activity: Hands-on coding activities performing file reading and writing in Java.	- Demonstrate an understanding of the importance of file handling in building real-world applications. - Write Java programs that use file input/output streams to read from and write to files using appropriate classes and exception handling. MAP: C02
16-17	Module 4– File Handling and Database Connectivity Lesson 2: Database Connectivity Lab Activity: Hands-on coding activities using JDBC to read and display data in Java.	At the end of the lesson, the students will be able to: - Explain the concept of database connectivity and describe how Java Database Connectivity (JDBC) is used to connect Java applications with relational databases - Show appreciation for the role of database connectivity in building dynamic applications. - Develop a simple Java program that connects to a database, retrieves, and displays it using JDBC. MAP: C02
18	Final Examination Covers Modules 3 and 4 Lecture and Laboratory Exam	Evaluate students' knowledge and skills relevant to all topics discussed in Modules 3 and 4.
References	1. Aguilar, R. (2024). <i>Object-oriented programming version 2.0</i> . Davao del Sur State College. 2. Burd, B. (2021). <i>Beginning programming with java for dummies</i> . John Wiley & Sons, Inc. 3. Budd, T.A. (2021). <i>An introduction to object-oriented programming</i> . Addison Wesley. 4. Davao del Sur State College. (2024). <i>Student handbook(1st ed.)</i> . 5. Sarcar, V. (2020). <i>Interactive object-oriented programming in java</i> . Springer Science. 6. 3G E-learning. (2023). <i>Object oriented programming</i> .	

TABLE OF CONTENTS

Course Overview	i
Preface	ii
Course Outline	iii
Table of Contents	ix
MODULE 1: INTRODUCTION TO OBJECT-ORIENTED PROGRAMMING...	1
Module 1 Lesson 1: Procedural vs Object-Oriented Programming	2
Module 1 Lesson 2: The Building Blocks of Java : A Review of Fundamentals Concept	6
Module Assessment	13
Module Summary	14
References	15
MODULE 2: OBJECT-ORIENTED PROGRAMMING IN JAVA.....	16
Module 2 Lesson 1: Objects and Classes	17
Module 2 Lesson 2: Encapsulation	26
Module Assessment	33
Module Summary	34
References	35
MODULE 3: OBJECT-ORIENTED DESIGN PRINCIPLES.....	36
Module 3 Lesson 1: Abstraction	37
Module 3 Lesson 2: Inheritance	44
Module 3 Lesson 3: Polymorphism	50
Module Assessment	59
Module Summary	60
References	61
MODULE 4: FILE HANDLING AND DATABASE CONNECTIVITY.....	62
Module 4 Lesson 1: File Input/Output Streams	63
Module 4 Lesson 2: Database Connectivity	71
Module Assessment	84
Module Summary	85
References	86

Appendices.....	87
Appendix A: Grammarian Certificate	88
Appendix B: Course Syllabus	89