# Course Outline

|  |  |
| --- | --- |
| **Course title: Object-Oriented Programming Design** | **Instructor name: Jun Albert Pardillo** |
| **Credit units: 3** | **Total hours: 54** |

## Course Description:

Object-Oriented Programming Design is a course designed for 2nd Year Electronic Engineering students who want to learn the fundamentals of object-oriented programming and design. This course will provide students with a comprehensive understanding of the principles of object-oriented programming, including encapsulation, inheritance, and polymorphism. The course will cover the basics of programming in Java, including data types, control structures, and object-oriented programming concepts. Students will learn how to design and implement object-oriented programs using Java, and will gain hands-on experience in developing software applications using object-oriented programming techniques. Throughout the course, students will work on a series of programming assignments and projects that will help them develop their programming skills and apply the concepts they have learned. They will also learn how to use software development tools such as Eclipse and Git, which are commonly used in the industry. By the end of the course, students will have a solid understanding of object-oriented programming and design, and will be able to apply these concepts to develop software applications in a variety of domains. They will also be well-prepared for more advanced courses in software engineering and computer science. Overall, Object-Oriented Programming Design is an essential course for any 2nd Year Electronic Engineering student who wants to develop their programming skills and gain a deeper understanding of object-oriented programming and design.

## Course Learning Outcomes (CLOs)

* Understand and apply the principles of object-oriented programming including encapsulation, inheritance, and polymorphism.
* Develop proficiency in Java programming, focusing on data types, control structures, and object-oriented concepts.
* Design and implement object-oriented programs using Java.
* Utilize software development tools such as Eclipse and Git for efficient programming and version control.
* Apply object-oriented programming techniques to solve real-world problems in various domains.

## Topics / Modules and Intended Learning Outcomes

1. Introduction to Object-Oriented Programming

* Describe the fundamental concepts of object-oriented programming and differentiate it from procedural programming paradigms.
* Explain the significance of encapsulation, inheritance, and polymorphism in object-oriented programming.

1. Java Programming Basics

* Write simple Java programs utilizing basic data types and control structures.
* Compile and execute Java programs, understanding the role of the Java Virtual Machine (JVM).

1. Encapsulation, Inheritance, and Polymorphism

* Implement encapsulation in Java programs to protect data and ensure robust code.
* Utilize inheritance and polymorphism to create flexible and reusable code.

1. Software Development Tools in Java

* Use Eclipse IDE for Java programming to enhance coding efficiency.
* Employ Git for version control in software development projects.

1. Object-Oriented Program Design and Implementation

* Design object-oriented programs by applying OOP principles effectively.
* Implement complex software applications using Java, demonstrating proficiency in object-oriented design and programming.

## Weekly Activities

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Week No.** | **Topic** | **Activity Description** | **Expected Output** | **Assessment Tools** |
| Week 1 | **Introduction to Object-Oriented Programming** | Lecture on the history and fundamentals of OOP. Introduction to Java programming environment setup. | Students will have a basic understanding of OOP concepts and setup their Java development environment. | Quiz on OOP concepts and practical assessment on environment setup. |
| Week 2 | **Java Programming Basics** | Hands-on session on writing simple Java programs covering basic data types and control structures. | Students will write and execute simple Java programs. | Code review and execution of written programs. |
| Weeks 3-4 | **Encapsulation, Inheritance, and Polymorphism** | Detailed lectures and lab sessions on implementing encapsulation, inheritance, and polymorphism in Java. | Students will implement classes using encapsulation, inheritance, and demonstrate polymorphism through various exercises. | Practical programming assignments and peer code reviews. |
| Weeks 5-6 | **Software Development Tools in Java** | Workshops on using Eclipse IDE for Java development and Git for version control. | Students will become proficient in using Eclipse and Git for their development projects. | Practical exercises and submission of a mini-project using Git. |
| Weeks 7-18 | **Object-Oriented Program Design and Implementation** | Series of lectures, project-based learning sessions, and peer reviews focusing on the design and implementation of object-oriented programs. | Students will design, implement, and present a software application using OOP principles. | Continuous assessment through project milestones, final project submission, and presentation. |

## References

*Streib, J. T., & Soma, T. (2023). Guide to java: A concise introduction to programming.*  
Link: https://books.google.com/books?hl=en&lr=&id=zQmtEAAAQBAJ&oi=fnd&pg=PR5&dq=%22Introduction+to+Object-Oriented+Programming%22+overview&ots=XhCdq0-RAJ&sig=vfQZyV26YLPlnUQxhiaxmjMeBfo

*Loy, M., Niemeyer, P., & Leuck, D. (2020). Learning Java: An Introduction to Real-World Programming with Java.*  
Link: https://books.google.com/books?hl=en&lr=&id=tsjZDwAAQBAJ&oi=fnd&pg=PP1&dq=%22Java+Programming%22+basics&ots=hLL6KLJZnU&sig=UrXZXHwWR80PYPn--y--3NWKziU

*Amantay, B. (2022). TEACHING MAIN CONCEPTS OF OBJECT-ORIENTED PROGRAMMING USING GREENFOOT INTEGRATED DEVELOPMENT ENVIRONMENT.*  
Link: https://journals.nauka-nanrk.kz/bulletin-science/article/view/3247

*Emmanuel, W. C., & Monden, A. (2021). Human Resource Analysis Based on Used Libraries in Eclipse Projects on GitHub.*  
Link: https://ieeexplore.ieee.org/abstract/document/9704958/

*Le, D. M. (2022). Model‐based automatic grading of object‐oriented programming assignments.*  
Link: https://onlinelibrary.wiley.com/doi/abs/10.1002/cae.22464