**Assignment #1**

**Full name: Arjun Thapa**

**Student-ID: M22W0385**

**There is no specific length requirement for your answer, but it is important to maintain clarity, consistency, and conciseness throughout your response.**

1- What is Object-Oriented Design, and how does it differ from other design methodologies?

Object-oriented design (OOD) is the process of planning a system of interacting objects for the purpose of solving a software problem[1]. The basic concepts of object-oriented design are Object or Class, Information hiding, Inheritance, Interface, and Polymorphism.

Object-Oriented Design (OOD) differs in several ways from other design methodologies like procedural design and functional design. Functional design is used to solve the problem of mathematical functions and procedural design is used to carry out specified duties through the procedures. Object-Oriented Design (OOD) usually focuses on the development of objects that combine data and functionality into a single entity which can communicate with other objects via clearly defined interfaces.

2- What is the Unified Modeling Language (UML), and how is it used in Object-Oriented Design?

Unified Modeling Language (UML) is a standardized modeling language consisting of an integrated set of diagrams, developed to help system and software developers for specifying, visualizing, constructing, and documenting the artifacts of software systems, as well as for business modeling and other non-software systems. The UML is an assortment of best engineering principles that have been effective in simulating huge, complicated systems[2].

UML is used in Object-Oriented Design to assist designers and developers in understanding, designing, and communicating a system's structure and behavior. Stakeholders may view and debate many parts of a system and make sure that everyone is on the same page with its design by utilizing UML diagrams. A system's design can be documented using UML, which makes it simpler to maintain, change, and evolve over time.

3- What are the different types of UML diagrams, and what is their purpose in Object-Oriented Design?

There are different types of UML diagrams which has each special purpose in Object Oriented Design. They are: -

1. Class Diagrams: - It shows the class, properties, and operations of the system. It helps in understanding the relationships between various classes and objects.
2. Object Diagrams: - It can be utilized to gauge how accurate class diagrams are. It symbolizes separate instances of classes and their interactions across time.
3. Use Case Diagrams: - Using characters and use cases, it simulates a system's functioning. It captures how a system must function and how actors are involved in that process. It depicts the system's use case perspective.
4. Sequence Diagrams: - By comparing the messages sent and received over time, it depicts how the items interacted. It defines the placement and hierarchy of an object's functions within a system.
5. Activity Diagrams: - It simulates how control moves across several activities. An activity diagram can be used to model both sequential and concurrent activities. It provides a visual representation of the workflow and the factors that influence events.

4- How can Object-Oriented Design be integrated with other software engineering practices such as Agile Development?

Object-Oriented Design be integrated with other software engineering practices such as Agile Development are: -

1. Iterative development: Iterative development, or agile development, involves creating software in brief, incremental cycles. Object-Oriented Design (OOD) is a good fit for this strategy since it offers a flexible and modular way to structure code. Developers can work on small, manageable portions of code in each iteration by dividing a system down into objects and classes.
2. Continuous integration: Continuous integration, which involves routinely integrating code changes into a shared repository, is another focus of agile development. Object-Oriented Design (OOD) supports this strategy by offering well defined interfaces between objects, which makes it simpler to incorporate code changes without the system breaking.
3. Test-driven development: Test-driven development, where tests are written before code is developed, is another component of agile development. By designing objects with clearly defined interfaces and testable behaviors, Object-Oriented Design (OOD) can be leveraged to create testable programming.
4. User stories and use cases: User stories and use cases are important tools in Agile Development for capturing and communicating user requirements. Using UML diagrams like use case diagrams and class diagrams, Object-Oriented Design (OOD) can be used to model these requirements, giving a visual representation of the system's architecture.
5. Refactoring: Refactoring, which entails regularly enhancing code to make it more scalable and manageable, is another aspect of agile development. Object-Oriented Design (OOD) is a natural way to restructure code since it allows for simple object-level organization and modification without affecting other parts of the system.

5- What do you hope to gain from this course?

I hope to gain a lot of knowledge from this course, like valuable information related to Object-Oriented Design and other related topics such as UML diagram, Design patterns, Agile Development system and other software systems. I also hope to get collaboration skill with collaborate with others student in group project.

REFERENCE: -

[1] 2023. Object-oriented design. *Wikipedia*. Retrieved April 28, 2023 from https://en.wikipedia.org/w/index.php?title=Object-oriented\_design&oldid=1151977805

[2] What is Unified Modeling Language (UML)? Retrieved April 28, 2023 from https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-uml/