

# Project 1 - part 1

**Name:** Vishal Prabhachandar

**Section:** CSCI 5448-001

1. Provide a definition of the OO term “class”

**Answer:**

A class in object-oriented programming acts as a blueprint for instantiating and managing one or more objects. It's a user-defined prototype that provides initial values to attributes for the state, methods for describing behavior, and constructors for initializing new objects.

---

2. Select any **three** of the following six OO terms:  
**abstraction, encapsulation, polymorphism**, coupling, cohesion, identity Provide:
  - a) A definition of the term
  - b) How the term applies to the OO notion of a class
  - c) What is a possible positive result of applying the term to a class design
  - d) What is a possible negative result of applying the term incorrectly in a class design

**Answer:**

Abstraction -

- a) Abstraction is the property of condensing the object to its essential details so that only the necessary characteristics are exposed to the end user. It is also a process of identifying the required characteristics and ignoring the irrelevant details of an object exposed.
- b) As abstraction shows only essential details and hides unnecessary information this reduces the complexity and enables user to implement complex logic in the class on top of the abstraction without worrying about the hidden information. An abstract class is blueprint definition of attributes and functions of a class that contain one or more abstract methods as well as concrete methods.

- c) The benefits of using abstraction to a class design are that it allows one to group related classes as siblings increasing software reusability and avoid code duplication. It also increases the security of the application as only essential details are exposed. The complexity of understanding code is reduced as abstract classes allow for writing shorter codes. Finally the code becomes more maintainable as changes can be done to internal code implementation without affecting other classes.
- d) The negative affects of applying abstraction to a class design incorrectly are that classes cannot be directly instantiated and in certain scenarios can add complexity where one needs to control situations and cases which are not required in the first place. The speed can also be impacted as the abstraction can make it slower instead of implementing it directly. In scenarios of constrained environments and small devices abstraction can lead to increase in code size and affect overall performance.

## Encapsulation -

- a) It is a process of bundling attributes and the functions that operate on the attributes into one single unit. Encapsulation also operates as a means to hide both variables and functions from being accessed by outside code.
- b) As encapsulation hides the internal state from external code and requires all the interaction to be performed through instantiated object's method thereby following the object oriented notions in a class.
- c) The advantages of applying encapsulation to a class design are that it makes code more flexible and maintainable thereby making it easier to change based on need and also debug when issues occur. It also enables reusing existing code by changing methods instead of building it from scratch. Encapsulation also offers data protection, preventing any external changes to data which might hinder the program from running.
- d) The drawbacks of using encapsulation in a class design incorrectly are that it can lead to increase time in program execution, as additional instructions are added in the code taking it longer time to complete. The code size increases as we need to provide specifiers for all the methods and have additional code to provide controlled access outside to the components.

## Polymorphism -

- a) It is the ability of methods, attributes and objects to take on multiple forms. It can also be put as the ability to access objects of various types using the same interface.
- b) Polymorphism presents an interface for several underlying data types and when a message is sent to different class objects, every object responds accordingly to the property of the class thereby following the object oriented notion.
- c) The positive result of applying polymorphism to a class design are that a single variable can be used to store different data types. It enables reusing code and provides means to override and overload classes. It also makes the code easier to debug in case of issues.

- d) The negative results of applying polymorphism incorrectly in a class design are that it can lead to degradation of performance where unnecessary as the decisions are taken during runtime to choose which function or variable to invoke and use. Polymorphism is also harder for the developers to implement and in turn can also lead to poor readability overtime.
- 

## References

- [Class OOP](#)
- [Class definition](#)
- [Abstraction](#)
- [Abstraction advantages and disadvantages](#)
- [Encapsulation](#)
- [Encapsulation advantages and disadvantages](#)
- [Polymorphism](#)
- [Polymorphism advantages and disadvantages](#)