DAC 0521 MET-M Writeup Java Module 1 Session - 03

Object Oriented Programming (OOP) - It is a programming methodology (an approach based on generally accepted principles) in which the implementation of a large software is divided into smaller units called called objects each containing its own state (data) and supporting its own behavior (operations).

Fundamental Goals

1: Data Abstraction - A class defines a set fields (variables) whose values indicate the state of an object of a particular type and a set of methods (functions) whose implementations describe the behavior of that object. It provides support for following mechanisms

(A) Activation - a new object (instance) is created from a class by first allocating a memory block for storing the values of the fields defined by that class and then initializing those values by calling a special method known as a constructor defined by that class.

(B) Binding - every object has a unique identity and when a method defined by a class is called upon its object the identity of this object is automatically passed (as an argument) to the implementation of that method.

(C) Containment - an object of one class exhibits ‘has a’ relationship with object of another class which it holds in one of its own fields. This relationship is called composition (Hotel has a Room) if the outer object controls the life-time of the inner object otherwise it is called aggregation (Room has a Guest).

2: Data Subtyping - A subclass (derived class) extends an existing superclass (base class) to define additional fields or methods or to override (provide new implementation for) its existing methods. It provides support for following mechanisms

(A) Dispatch - a method defined by a class can be called on an object of its subclass and if this subclass has overriden that method then the implementation provided by the subclass will be invoked.

(B) Interface - an abstract (non-activatable) type can define pure (unimplemented) methods which must be overridden by its activatable subclasses.

(C) Inheritance - an object of a subclass exhibits ‘is a’ relationship with its superclass. This relationship is called specialization (Sports-Car is a Car)if the superclass is activatable (non-abstract) otherwise it is called realization (Car is a Vehicle).