Java FAT prep – All definitions

Name: Jonathan Rufus Samuel – 20BCT0332

**Module - 1**

**Class:** A class is a user defined blueprint or prototype from which objects are created.

**Object**: It is a basic unit of Object-Oriented Programming and represents the real life entities. A typical Java program creates many objects, which as you know, interact by invoking methods.

**Inheritance**: Inheritance in Java is a mechanism in which one object acquires all the properties and behaviours of a parent object. It is an important part of OOPs (Object Oriented programming system). The idea behind inheritance in Java is that you can create new classes that are built upon existing classes. When you inherit from an existing class, you can reuse methods and fields of the parent class. Moreover, you can add new methods and fields in your current class also. Inheritance represents the IS-A relationship which is also known as a parent-child relationship. Used for Method Overriding and Code Reusability.

**Polymorphism**: Polymorphism means "many forms", and it occurs when we have many classes that are related to each other by inheritance. Like we specified in the previous chapter; Inheritance lets us inherit attributes and methods from another class. Polymorphism uses those methods to perform different tasks.

**Encapsulation**: Encapsulation in Java is a mechanism of wrapping the data (variables) and code acting on the data (methods) together as a single unit. In encapsulation, the variables of a class will be hidden from other classes, and can be accessed only through the methods of their current class. Therefore, it is also known as data hiding.

**Abstraction**: Abstraction is a process of hiding the implementation details and showing only functionality to the user. Another way, it shows only essential things to the user and hides the internal details, for example, sending SMS where you type the text and send the message. You don't know the internal processing about the message delivery.

**Dynamic** **Binding**: When type of the object is determined at run-time, it is known as Dynamic Binding.

**Static Binding:** When type of the object is determined at compilation time.

**Concurrency**: Concurrency is the ability to run several programs or several parts of a program in parallel. If a time-consuming task can be performed asynchronously or in parallel, this improves the throughput and the interactivity of the program. A modern computer has several CPU’s or several cores within one CPU. The ability to leverage these multi-cores can be the key for a successful high-volume application.

**Other Features of Java:** Platform Independent, Robust, Secure, Portability, Memory Efficient and Concurrent. Also Object Oriented.

**Strongly Typed**: Java is strongly typed; every type is strictly defined and every variable and expression has a type.

**JVM:** JVM (Java Virtual Machine) is an abstract machine. It is a specification that provides runtime environment in which java bytecode can be executed. JVM (Java Virtual Machine) is an abstract machine. It is a specification that provides runtime environment in which java bytecode can be executed.

**Why do we use an Interface?**

It is used to achieve total abstraction. Since java does not support multiple inheritances in the case of class, by using an interface it can achieve multiple inheritances. It is also used to achieve loose coupling. Interfaces are used to implement abstraction. So, the question arises why use interfaces when we have abstract classes? The reason is, abstract classes may contain non-final variables, whereas variables in the interface are final, public and static.

**Module – 2**