**71.Explain the difference among structure and a class?**

## Ans: Class

* It is defined using ‘class’ keyword.
* When data is defined in a class, it is stored in memory as a reference.
* It gets memory allocated only when an object of that class is created.
* The reference type (before creating an object) is allocated on heap memory.
* They can have constructors and destructors.
* It can use inheritance to inherit properties from base class.
* The ‘protected’ access modifier can be used with the data members defined inside the class.

**Structure**

* The ‘struct’ keyword is used to define a structure.
* Every member in the structure is provided with a unique memory location.
* When the value of one data member is changed, it doesn’t affect other data members in structure.
* Total size of the structure is equivalent to the sum of the size of every data member.
* It is used to store various data types.
* It takes memory for every member which is present within the structure.
* Its instance can be created without a keyword.
* It doesn’t support protected access modifier.
* It doesn’t support inheritance.
* It doesn’t have a constructor or destructor.
* The values allocated to structures are stored in stack memory.

**72. Explain the default access modifier in a class?**

There are four types of Java access modifiers:

1. **Private**: The access level of a private modifier is only within the class. It cannot be accessed from outside the class.
2. **Default**: The access level of a default modifier is only within the package. It cannot be accessed from outside the package. If you do not specify any access level, it will be the default.
3. **Protected**: The access level of a protected modifier is within the package and outside the package through child class. If you do not make the child class, it cannot be accessed from outside the package.
4. **Public**: The access level of a public modifier is everywhere. It can be accessed from within the class, outside the class, within the package and outside the package.

**73. Can you list out the different types of constructors?**

* **No-argument constructor:**A constructor that has no parameter is known as the No-argument or Zero argument constructor. If we don’t define a constructor in a class, then the compiler creates a **constructor(with no arguments)** for the class. And if we write a constructor with arguments or no arguments then the compiler does not create a default constructor.
* ***Note*: Default constructor provides the default values to the object like 0, null, etc. depending on the type**.
* **Parameterized Constructor:**A constructor that has parameters is known as parameterized constructor. If we want to initialize fields of the class with our own values, then use a parameterized constructor.
* **Default Constructor**: A constructor that has no parameters is known as default the constructor. A default constructor is invisible. It is taken out.It is being overloaded and called a parameterized constructor. The default constructor changed into the parameterized constructor. But Parameterized constructor can’t change the default constructor.

**74**. **Explain a ternary operator?**

In Java, the **ternary operator** is a type of Java conditional operator. The meaning of **ternary** is composed of three parts. The **ternary operator (? :)** consists of three operands. It is used to evaluate Boolean expressions. The above statement states that if the condition returns **true, expression1** gets executed, else the **expression2** gets executed and the final result stored in a variable.

**Example:**

**{**int n1 = 5, n2 = 10, max;

        System.out.println("First num: " + n1);

        System.out.println("Second num: " + n2);

        // Largest among n1 and n2

        max = (n1 > n2) ? n1 : n2; **}**

**75. Do We Require Parameter For Constructors?**

**Default Constructor –**A constructor that accepts no parameter is called Default Constructor. It is not necessary to have a constructor block in your class definition. If you don't explicitly write a constructor, the compiler automatically inserts one for you.

**76. Explain Sealed Modifiers?**

When applied to a class, the sealed modifier prevents other classes from inheriting from it. In the following example, class B inherits from class A , but no class can inherit from class B . You can also use the sealed modifier on a method or property that overrides a virtual method or property in a base class.

**77. Explain The Difference Between New And Override.**

override: overrides the functionality of a virtual method in a base class, providing different functionality. new: hides the original method (which doesn't have to be virtual), providing different functionality. This should only be used where it is absolutely necessary.

**78. How Can We Call The Base Method Without Creating An Instance?**

Static methods are the methods in Java that can be called without creating an object of class. They are referenced by the **class name itself** or reference to the Object of that class.

**79. Define The Various Types Of Constructors?**

* **No-argument constructor:**A constructor that has no parameter is known as the No-argument or Zero argument constructor. If we don’t define a constructor in a class, then the compiler creates a **constructor(with no arguments)** for the class. And if we write a constructor with arguments or no arguments then the compiler does not create a default constructor.
* ***Note*: Default constructor provides the default values to the object like 0, null, etc. depending on the type**.
* **Parameterized Constructor:**A constructor that has parameters is known as parameterized constructor. If we want to initialize fields of the class with our own values, then use a parameterized constructor.
* **Default Constructor**: A constructor that has no parameters is known as default the constructor. A default constructor is invisible. It is taken out.It is being overloaded and called a parameterized constructor. The default constructor changed into the parameterized constructor. But Parameterized constructor can’t change the default constructor.

**80.** **Define Manipulators?**

A Java manipulator is your own code in Java that takes records from any number of pipeline components in Forge or, optionally, your source data, and changes it according to your. processing requirements. A Java manipulator can then write any records you choose to its output