Q: What is JAVA?

Java is an object oriented programming language. It is also highly structured, platform independent and widely use programming language. Its develop by sun microsystem but now owned by oracle

Q: Why you choose java?

I choose to learn java because its platform Independent, its object oriented, its support web based, Robust, secured and support multithread.

Q: why java is a OOP?

Java is OOP because it follows four concepts. Those concepts are abstraction, encapsulation, inheritance and polymorphism. Those are create working variables and methods then re-use all or part of them use without compromising security.

Q: Describe all supported Java concepts?

Java Support following four for concepts:

1. Abstraction.
2. Encapsulation.
3. Inheritance.
4. Polymorphism.

**Abstraction:** Data Abstraction is a process of hiding certain details and showing only essential information of the user. Abstraction can be work with either abstract class or interface.

**Encapsulation**: Encapsulation in Java is a mechanism of wrapping the data (variables)

and code acting on the data (methods) together as a single unit. Declare the variables of a class as private. Provide public setter and getter methods to modify and view the variables values.

**Inheritance**: inheritance is a main pillar of OOP.Inheritance can be defined as the process where one class acquires the properties (methods and fields) of another. The class which inherits the properties of other is known as subclass (derived class, child class) and the class whose properties are inherited is known as superclass (base class, parent class).

**Polymorphism**: poli means many and morph means forms. So simply the process is representing one from to multiple from is known as polymorphism. Method overloading is compile time polymorphism and overwriting is runtime polymorphism.

Q: What is Method Overloading?

when we create more than one method in same class, same name with different parameter is called method Overloading.

Q: what is method Overwriting?

when we create same method name, same parameter, same return type in parents & child class is called method overwriting.

Q: Why JAVA not a pure OOP?

Java not a pure OOP because java use primitive datatype and java cannot inherit more than one class at a time.

Q: What is Package?

A package is a collection of related classes and intrerface.

**Q: What is Class?**

Class is a simply representation or collection of object. In other’s we tell that it’s a blueprint of object or collection of variables, method, constructor and blocks.

Q: What is different between inner class and nested class?

Nested class are two categories. One is static and another one non-static. When nested class declare with static keyword is called static nested class and when nested class declare without static key word is called inner class.

**Q: What is Object in Java?**

In simply we tell that object is an Instance of class.it means object is a piece of code that represents real life entity.it has own state, behavior and identity’s .it contains reals’ values instead of variable. Without class object does not exist.

**Q: What is method?**

A method is a collection of statement that are grouped together to perform and operation.

**Q: What is variable?**

A variable is an identifier or reference of value. it also called name of memory location.

**Q: How many types of variable? describe all of them**.

There are Three types of variables in Java. These are given bellow:

1. Local variable
2. Global or instance variable
3. Static or class variable.

**Local variable**: This variable declares inside the method, constructor or block. We use this variable only those specific areas not outside of area. In local variable never declare with static key word.

**Global or instance variable:** This variable declared outside the method, constructor or blocks but inside the class. we use this variables not only this class also use whole project by using create object or inheritance those class.

**Static or class Variable**: when a variable declared with static keyword in class level its called Static or class Variable. We can access it just with class name anywhere in project.

Q: What is constructor? Describe constructor rules.

A constructor is a special type of method that use initialize the object.in other way constrictor is a block of code which is create the time of execution.

Rules of Constructor are given bellow:

> Constructor name must be class name.

> Constructor must have no return type not even void.

> It is called automatically at the time of object creation.

> Default constructor (no parameter) or Parameterize contractor.

> Constructor van be overloaded but cannot overridden.

Q: what is data type?

Data types specify the different sizes and values that can be stored in the variable. It also called space in the memory. There are two types of data types in Java:

1. Primitive datatype: 8 types of primitive datatype.

|  |  |  |  |
| --- | --- | --- | --- |
| Use Full Number | Use decimal number | Use single char | True/False |
| Byte- 1 byte | double-8 byte | char 2 bytes | boolean I byte |
| short- 2bytes | float -4bytes |  |  |
| int- 4 bytes |  |  |  |
| long – 8 bytes |  |  |  |

1. Non Primitive data type: Non-primitive data types are called **reference types** because they refer to objects

Q: Describe different between primitive and non-primitive datatype?

* Primitive types are predefined (already defined) in Java. Non-primitive types are created by the programmer and is not defined by Java (except for String).
* Non-primitive types can be used to call methods to perform certain operations, while primitive types cannot.
* A primitive type has always a value, while non-primitive types can be null.
* A primitive type starts with a lowercase letter, while non-primitive types starts with an uppercase letter.
* The size of a primitive type depends on the data type, while non-primitive types have all the same size.

Q: What is interface?

An **interface in Java** is a blueprint of a class. It contains static constants and

abstract methods. In other word we say that interface is a collection of abstract method or unimplemented method and variables.

Q: Why we use JAVA interface?

We use java interface mainly three reasons to use interface. The reasons are:

1. It is used to achieve abstraction.
2. By interface, we can support the functionality of multiple inheritance.
3. It can be used to achieve loose coupling.

Q: can you describe relationship between classes and interface?

A class extends another class, an interface extends another interface, but a **class implements an interface**.

Q: What is Abstract class?

A class which is declared with the abstract keyword is known as an abstract class in Java. It can have abstract and non-abstract methods (method with the body).

Point of remainder of Abstract class:

* An abstract class must be declared with an abstract keyword.
* It can have abstract and non-abstract methods.
* It cannot be instantiated.
* It can have [constructors](https://www.javatpoint.com/java-constructor) and static methods also.
* It can have final methods which will force the subclass not to change the body of the method

Q: What is modifier? Describe all of them.

A modifier is a keyword that placed in a class, variable and method declaration to change their access level how to operate it.

Two types of modifier.

1. Access modifier:
2. Non Access Modifier.

Access Modifier: Four types of Access modifier.

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.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name of Access  Modifier | Same Package | | Other Package | |
| By obj create | By inherits | By obj create | By inherits |
| Default | Yes | Yes | No | No |
| Public | Yes | Yes | Yes | Yes |
| Protected | Yes | Yes | No | Yes |
| Private | No | No | No | No |
| In top class only Public and Default can be use & private and protected not accessible. | | | | |

Non access Modifier: In java there are 7 types of non-access modifier. They are used with classes, methods, variables, constructors etc to provide information about their behavior to JVM. They are:

1. static 2. Final 3. Abstract 4. Synchronized

5. Transient 6.Volatile 7. Native