**1 – Why we need to use OOP ? Some major OOP languages ?**

OOP is transferring real-life objects to computer, OOP sets 6 priority goals for software development.

1-Natural : through this, program becomes understandable.

2-Reliable : Well-designed, carefully-written program is reliable.

3-Reusable : OO classes are reusable, so programmer doesn't have to write again same codes.

4-Maintainable : when we need to update a feature from the program, we don't write program again, small code updates help us.

5-Extendable : like maintainability, we may need to extend to the program, oop provides extendability to us with inheritance, polymorphism, overriding,delegation and design patterns etc.

6-Timely : we can work with each code separately, even we can work with more than one programmer for same problem. And this saves time.

Java,C#,C++,PHP,Phyton,Objective-C, Smalltalk, Delphi, Swift, Perl, Ruby are some of major OOP languages.

**2 – Interface vs Abstract class ?**

More than one Interface can be implemented one class, but only one abstract class can be implemented one class.

In Interfaces, methods creates without body, but in abstract classes, methods can create with body or without body, because of that when a Interface is implemented a class, methods of the Interface must override in the class.

Interfaces does not have access modifiers, all objects are assumed public modifier.

Interfaces aren't contain constructor.

Methods of Interfaces can't be static. Methods of abstract class can be static if methods are not abstract.

**3 – Why we need equals and hashcode ? When to override ?**

Equals compare two objects and check if these objects are equal.There are 2 ways for comparation :

Shallow comparison : By default, equals checks if two objects are same (x==y).

Deep comparison : We can override equals method for defining what equality means our objects.

Hashcode method returns integer value based on results of equals method.When the equals method returns true for two objects, the hashcode method returns same hash value for these objects.

Otherwise, returns different hash values for these object.

If equals method is overridden, hashcode method must also be overridden.

**4 – Diamond problem in Java ? How to fix it?**

Assume that we have 4 classes: A,B,C,D. A is a superclass and implemented B and C. Then when B,C implemented D.

If B and C took a method from A and this method overrode it in both classes, then if we want to use this method in D, D will take the method from which class, B or C ?

Here is, Diamond problem. But in Java (up to version 8), diamond problem doesn't exist because multiple inheritance is provided by Interfaces.

If we use a method from Interface, the method is overridden every class we use.

After version 8, Interfaces can have methods with body, but this time java compiler inhibits Diamond Problem.

So, if we cause Diamond Problem by using the method in D as i mentioned in the line above, compiler throws a exception and warns us to fix the code.

**5 – Why we need Garbage Collector ? How does it run ?**

Garbage Collector do the management of memory for us, looks into Heap, finds used objects and delete the unused objects.

**6 – Java ‘static’ keyword usage ?**

The static can be variable,method,block,class.

When a variable is static, they are created at class level,i.e the static variable is in the same memory location as own class.

All instances of the class uses same static variables.

Also, we can use static variables without creating a object.

Like static variables, static methods are also at class level and able to use without creating a object.

In static methods, we can not call unstatic methods and variables.

Static blocks are used to initialize static variables.

Only nested classes can be made static.And nested class does not need reference of outer class.

**7 – Immutability means ? Where, How and Why to use it ?**

Immutability means unchangeable. Wrapper classes like Integer,String,Boolean are Immutable classes, which means they are created once and can not be changed.

We can use Immutability for protect our objects.

If we want create a Immutable object, we create class as final,all variables and changeable fields as final and private. For private variables, do not add setter.

**8 – Composition and Aggregation means and differences ?**

Composition and Aggregation are methods using for creating a relationship between two objects.

For example, a company has various units. These units belong to company and belong only a company.

There are workers in these units.Also, there is a relationship between units and workers.

If the company shuts down, existence of the units ends. But existence of the workers continues, also the workers can work another unit or company before the company shuts down.

Based on this example, Composition is the relationship between the company and the units and Aggregation is the relationship between the workers and the units.

And Composition is stronger relationship than Aggregation.

**9 – Cohesion and Coupling means and differences ?**

Coupling represent the relationship between classes. When two classes are dependent on each other and we changed something in one class, the other class also will be affected.

This situation is not wanted in OOP because of maintenance costs and modularity.

Cohesion represent the relationship between objects of class. Because of single responsibility rule in OOP,related objects should be in one class.

According to the explanations above, loose coupling and high cohesion are wanted in OOP.

**10 - Heap and Stack means and differences ?**

Stack and Heap are logical structures where stored in the computer's RAM.

Heap is used for datas of uncertain size. Otherwise Stack is used.

Using Stack is easier than using Heap.

Reaching Stack is faster than Heap.

Variables are used with pointer in Heap, there are no necessity like that in Stack.

Variables cannot be resized in Stack, in heap, variables can be resized.

In Stack, Allocation and Deallocation are done by compiler instructions automatically, in Heap, Allocation and Deallocation are done by programmer manually.

**11 – Exception means ? Type of Exceptions ?**

Exception means unusual situations in the program. If occurs a unusual situation, compiler creates a exception object and this object contains informations about the unusual situation/error.

There are 3 types of Exception.

Checked Exception : When we write a code block, program wants to create try-catch block or throws explanations from us, this is checked exception.

UnChecked Exception : This type of exception, program doesn't force us to create a try-catch or throws. We have to find solution for this exception.

Error : When serious problem occurs in the program, it is called Error.

**12 – How to summarize ‘clean code’ as short as possible ?**

If the code is easily understandable, readable and not including repeating parts,this code is clean code.

**13 - What is the method of hiding in Java ?**

When a static method is written in the child class with the same signature in the parent class, the method of child class hides the method of parent class.

It means now,the method of child class is valid and if the method is called, run the code in the method of child class.

**14 - What is the difference between abstraction and polymorphism in Java ?**

Polymorphism means the same object that can take multiple forms. Assume that, we have classes called Animal, Cat and Dog. When Animal class implemented Dog and Cat and we want to create a object with Animal class,

this object can be from Cat or Dog class.

Abstraction helps us for Polymorphism. Interfaces and abstract classes are used for abstraction. Again, we have classes called Animal, Cat and Dog. For example, animals have sounds.If Dog class is implemented from

Animal class and we create a sound method in Animal class, we can use the method in Dog class and override if we want.