

1) What is difference between C++ & Java?

JAVA

C++

- | | |
|--|---|
| 1) Java doesn't support pointers | 1) It supports pointer concept. |
| 2) It doesn't support multiple inheritance. | 2) It supports multiple inheritance. |
| 3) It doesn't include structure & unions | 3) It has structure & union concept. |
| 4) Java includes automatic garbage collection. | 4) C++ requires explicit memory management. |
| 5) Java has method overloading but no operator overloading. | 5) C++ supports both, method overloading as well as operator overloading. |
| 6) It is a platform independent programming language. | 6) It is platform dependent programming language. |
| 7) Java has compiler and interpreter both. | 7) C++ uses only compiler. |
| 8) Used for web based application but also used for developing desktop applications. | 8) Used for design only desktop applications like OS, compiler etc. |

2) What are key features of java?

- 1) Object Oriented
- 2) platform independent
- 3) Simple
- 4) Secure
- 5) Architecture - neutral
- 6) portable
- 7) Robust
- 8) multithreaded
- 9) Interpreted
- 10) High performance.

3) What is JVM?

- Java Virtual machine is a specification that provides runtime environment in which java bytecode (.classfile) can be executed. The JVM is a platform independent as the name implies. The JVM acts as a 'virtual' m/c or processor. Java is platform independent consists mostly of its java virtual m/c. JVM makes this possible because it is aware of the specific instruction lengths & other particularities of the platform.
- The JVM is not platform independent - JVM provides the environment to execute java file. So at the end it depends on your kernel & kernel is differ from os to os. The JVM is used to both translate the bytecode into the m/c understandable code for a particular computer & actually execute the corresponding m/c language instructions as well. Without JVM you can not run your java application.

4) What is oops? why java called as object oriented?

→ oops stands for Object Oriented programming.
It has several advantages:

- 1) oop is faster and easier to execute
- 2) oop provides a clear structure for the programs
- 3) oop helps to keep the java code DRY 'Don't repeat yourself', and makes the code easier to maintain, modify & debug.

• Java is purely an object oriented language due to the absence of global scope, Everything in java is an object.

• All the program codes & data resides within classes of objects. It comes with an extensive set of classes, arranged in package, object model in java is simple and easy to extend.

5) what is multiple inheritance? Explain with real time & in a programmatic way?

→ multiple inheritance is a feature of some object-oriented computer programming languages in which an object or class can inherit characteristics and features from more than one parent object or parent class.

• It is distinct from single inheritance, where an object or class may only inherit from one particular object.

• The problem occurs when there exist methods with same signature in both the super classes & subclass. On calling the method, the compiler can not determine

which class method to be called and even on calling
which class method gets the priority.

The reason why Java doesn't support multiple inheritance

class Parent1 {

void fun() { }

class Parent2 {

void func() { }

class Parent3 extends Parent1, Parent2 {

public static void main(String args[]) {

Parent3 three = new Parent3();

three.fun();

}

8) Abstract Class

Interface

- 1) Abstract class can have abstract and non-abstract methods. Interface can have abstract methods since Java 8, default, static.
- 2) Abstract class doesn't support multiple inheritance. Interface supports multiple inheritance.
- 3) Abstract class can have final. Interface has only static & final non-final, static, non-static variables. Variables.
- 4) Abstract class can provide the implementation of interface. Interface can't provide the implementation of abstract class.
- 5) The abstract keyword is used to declare abstract class. The interface keyword is used to declare interface.
- 6) A java abstract class can have members like private, protected etc. members of a java interface are public by default.
- 7) An abstract class can extend another java class & implement multiple java interface. An interface can be implemented extend another java interface only.
- 8)

```
public abstract class Shape {  
    public abstract void draw();  
}
```

```
public interface Drawable {  
    void draw();  
}
```

8) what is polymorphism and inheritance?

- Polymorphism means ability to perform one action in different forms.
- we achieve polymorphism using method overloading & method overriding
- method overloading also known as Static polymorphism or Compile time polymorphism or early binding.
- method overriding also known as Runtime polymorphism or late binding
- Inheritance means child class can acquire parent class properties and also it can have its own properties.
- For method overriding we require inheritance.
- It is useful for code reusability, reuse attributes & methods of an existing class when you create a new class.

9) what is Collection? what's it's need? if we are having array?

Arrays

Collections

1) Arrays are fixed in size Collection are growable in nature that we cannot change further. that is based on our requirement

2) with respect to memory arrays with respect to memory collection are not recommended to use. are recommended to use.

3) with respect to performance with respect to performance arrays are recommended collection are not recommended to use.

- 4) Arrays can hold only homogeneous data types elements
Collection can hold both homogeneous & heterogeneous elements.
- 5) There is no underlying data structure for arrays & hence based on some standard data ready made method support structure & hence for every requirement ready made method support is available being a performance.
- 6) Arrays can hold both object and primitive.
Collection can hold only object type but primitive.

17

Arrays

ArrayList

- 1) An array is a dynamic created object. It serves as container that holds the constant number of values of the same type. It has contiguous memory location. The ArrayList is a class of Java collection framework. It contains popular classes like Vector, Hashtable etc.
- 2) Array is static in size. ArrayList is dynamic in size.
- 3) An array is a fixed-length data structure. ArrayList is a variable-length data structure. It can be resizable itself whenever needed.
- 4) Array can be multi-dimensional. ArrayList is always single-dimensional.

- 5) It is mandatory to provide the size of an array while initializing ArrayList without specifying it directly or indirectly. we can create an instance of ArrayList by the arrays in java. The size T once creates an ArrayList of default size.
- 6) It performs fast in comparison to ArrayList because of fixed size. ArrayList is internally backed by the arrays in java. The resize operation in ArrayList turns down the performance.

12) Difference between List and ArrayList.

List

ArrayList

- | | |
|--|---|
| 1) List is an interface. | ArrayList is a class. |
| 2) List interface extends the Collection framework. | ArrayList extends AbstractList class & implements List interface. |
| 3) List cannot be instantiated. | ArrayList can be instantiated. |
| 4) List interface is used to create a list of elements which are associated with their index number. | ArrayList class is used to create a dynamic array that contains object. |
| 5) List Interface creates a collection of elements that are stored in sequence and they are identified and accessed using the index. | ArrayList creates an array of objects where the array can grow dynamically. |

13) Explain Exception handling & how it is handled?

- · Exception handling ensures that the flow of the program doesn't break when an exception occurs.
- For example, if a program has bunch of statements and an exception and an exception occurs mid-way after executing certain statements then the statements after the exception will not execute and the program will terminate abruptly.
- By handling we make sure that all the statements execute and the flow of program doesn't break.
- We can handle exception by using 5 keywords.
1) try 2) catch 3) finally 4) Throw 5) Throws.
- We write our essential code in finally block whether there is an exception or not.

14) What is Garbage Collection?

- · In Java, garbage means unreferenced objects.
- Garbage collection is process of reclaiming the runtime unused memory automatically. In other words, it is a way to destroy the unused objects.
- To do so, we were using free() function in C language & delete() in C++. But, in java it is performed automatically. So, java provides better memory management.

Advantages:-

- It makes java memory efficient because GC removes the unreferenced objects from heap memory.
- It is automatically done by the GC (part of JVM) so we don't need to make extra efforts.

• Three ways to object be unreferenced:

i) By nulling the reference.

ii) By assigning a reference to another

iii) By anonymous object etc.

Q) Explain public static void main?

i) **public** :- It is an access modifier, which specifies from where and who can access the method.

• making main method public makes it globally available.

• It is made public so that JVM can invoke it from outside the class as it is not present in the current class.

ii) **Static** :- It is a keyword which is when associated with a method, makes it a class related method.

• The main() method is static so that JVM can invoke it without instantiating the class.

• This also saves the unnecessary wastage of memory which would have been used by the object declared only for calling the main() method by the JVM.

iii) **void** :- It is a keyword & used to specify that a method doesn't return anything. As main() method doesn't return anything, its return type is void.

iv) **main** :- It is the main name of java main method. It is the identifier that the JVM looks for as the starting point of the java program. It's not a keyword.

v) `String[] args` :- It stores java command line arguments and is an array of type `java.lang.String` class. Here, the name of the string array is `args` but it is not fixed and user can use any name in place of it.

17) Write a program to reverse a String?

Class Reverse {

Logic part :- public ~~String~~ void reverse (String s) {
 String str = "";
 for (int i = s.length () - 1; i >= 0; i--) {
 str += s.charAt (i);
 }
~~return str;~~
 System.out.println ("str");

public static void main (String [] args) {
 Reverse R = new Reverse (~~args~~);
 R.reverse ("Shivtej");

18) Write a program to check whether two strings are anagram or not?

import java.util.Arrays ;

public class AnagramString {

static void isAnagram (String str1, String str2) {
 String s1 = str1.replaceAll ("\\s", "");
 String s2 = str1.replaceAll ("\\s", "");
 boolean status = true;

```
if (s1.length() != s2.length()) {  
    status = false;  
}  
else {  
    char[] ArrayS1 = s1.toLowerCase().toCharArray();  
    char[] ArrayS2 = s2.toLowerCase().toCharArray();  
    Arrays.sort(ArrayS1);  
    Arrays.sort(ArrayS2);  
    status = Arrays.equals(ArrayS1, ArrayS2);  
}  
  
if (status) {  
    System.out.println(s1 + " and " + s2 + " are anagrams");  
} else {  
    System.out.println(s1 + " and " + s2 + " are not anagrams");  
}  
  
public static void main(String[] args) {  
    isAnagram("keep", "peek");  
    isAnagram("mother in law", "Hitler Woman");  
}
```

output : keep and peek are anagrams
Mother In Law & HitlerWomen are anagrams.

2) What is REST API?

- REST or RESTful API design (Representational State transfer) is designed to take advantage of existing protocols.
- While REST can be used over nearly any protocol, it usually takes advantage of ~~existing protocols~~ while HTTP when used for web APIs.
- We don't have to install libraries or additional software in order to take advantages of a REST API.
- It is to be notable that for its incredible layer of flexibility.
- Since data and is not tied to methods of resources, REST has the ability to handle multiple types of calls, return different data formats & even change Structure with the correct implementation of hypermedia.

7) # Encapsulation = data hiding + Abstraction

(mob. versions) phones
android 6, 7, 8, 9

Class Account

{

data
hiding

→ private double balance;

public double getBalance()
{ // validate

return balance;

}

public void setBalance(double amount)

{

// validate

this.balance = this.balance + amount;

}

Advantages:-

- 1) Security → when security is necessary then only go for encapsulation
- 2) Maintainability } abstraction
- 3) Modularity }
- 4) Enhancement

Disadvantage:-

- 1) low performance (getter setters) time consuming validations
- 2) length of code is more because of validations.

9) # method Overloading :- Same method name, different parameters.

- adv:-
 - 1) complexity reduce
 - 2) more flexibility.

Compiler (method resolution based on reference type) Hence known as compile type polymorphism or static polymorphism.
Early binding.

Test t = new Test();

test type t runtime type

Method overriding :-

• child class is not satisfied with parent class method, that method child class can implement.

Same name, same ~~parameters~~ Parameter, Same sequence.

JVM is responsible to call overridden method. (method resolution is on runtime object)

Therefore known as runtime Polymorphism, Dynamic binding, ^{Late} binding

P p = new P();

P.marry() → parent class

C c = new C();

C.marry() → child class

P ^{o type}

P p₁ = new C();

p₁.marry() → child class

P p₁ = new C();

p₁.fun()

child class (all function only)

• P is P type so compiler will check if marry() is present in P class otherwise give compile time error.

• At runtime JVM will check that method is overridden in child class or not.

Rules : 1) method signature must be same, Name & argument.

2) co-variant return types are allowed after L4 now
Same or its child type allowed after L5 V

3) covariant return types for wrapper types not
for primitives.

4) for private methods are not possible in overriding.
if do so, then its independent methods