

1: What is an object in java?

- Object Is an Instance of class and it has its own state and behavior.
- In real world we can say, Dog is object of Animal class which have different state like breed, color, name, hungry, etc and behavior like wagging tail, fetching, barking etc.

2: What is the class?

- A class is the blueprint or we can say template from which individual objects are created.

3: What is a constructor?

- Constructor is a code block just like a method which is used to initialize the state of an object.
- It will be invoked at the time of object creation to construct the value for object.

4 : What is default constructor?

- If there is not any constructor in class then java compiler creates default constructor.
- It is no argument constructor which initializes any uninitialized fields to their default values.

5 : Constructor returns any value?

- Yes. It will return instance of current class (However we can not use return type with constructor).

6 : What is the difference between static and not static variable?

- Static variables are preceded by static keyword. For non-static variable, there is not any preceding keyword.
- Memory is allocated for static variables at the time of class loading. Memory is allocated to non- static variables whenever an object is created.
- Memory is allocated only once to static variables on class loading. Memory is allocated multiple time whenever a new object is created to non-static variables.
- Static variable example : Collage name of students, Company name of employees..

7 : What is the difference between static and not static(Instance) method?

- Method declared with static keyword is static method. If Method declared without static keyword then it is instance method.
- No need of object to call static methods. Object needed to call instance method.
- Can not access non static stuff inside static methods directly. Opposite to it, We can access static and non static stuff directly inside instance method.

8 : What is inheritance in java?

- In Java, Inheritance provides mechanism using which one object of child class can acquire all the properties and behaviors of parent object.
- It will crate IS-A relationship.
- Main usage of inheritance in java is for code re-usability and method overriding to achieve run-time polymorphism.

9 : Multiple inheritance is supported in java on class level? If No.. Why?

- No.. Multiple inheritance is not supported in java in case of class to simplify the language and reduce the complexity.

10 : What is method overriding in java?

- Method overriding is a feature which allows a child class or sub class to provide a specific implementation of a method which is already provided by one of its parent classes or super class.
- It is used for runtime polymorphism.

11 : Why main method is static?

- As we know, We can access static stuff without creating object of class. Because of static keyword with main method, Java virtual machine can directly call it without creating object of class.
- This way it will provide kind of root to start execution of program.

12 : What is method overloading?

- Method overloading is ability to create multiple methods with same in same class but with different signatures (different input parameters and types).
- Method names will be same but parameters will be different for all overloaded methods.

13 : What is constructor overloading?

- Same as method overloading, Single class can have multiple constructors with same name as class name but all have different signatures (different input parameters and types) is called constructor overloading.

14 : Can we override static methods?

- We can declare static method with same signature in subclass but it will not behave as overridden method. So answer is No.
- We can not override static methods as they are part of class not object.
- You can override static methods but output will be different than the expected.

15 : How to reverse string in java?

- **StringBuffer** class has a method called **reverse()**. We can use it to reverse the string.

Example :

```
1 public static void main(String[] args) {  
2  
3 // buffer string using StringBuffer class.  
4  
5 StringBuffer a = new StringBuffer("I like java very much.");  
6  
7 // use reverse() method to reverse string  
8  
9 System.out.println(a.reverse());  
10  
11 }
```

16 : Can we overload static methods?

- Yes.. There is not any restriction to overload static methods.
- We can overload static and non static methods in java.

17 : Can we use private member of parent class in sub class?

- No.. It will not allow to use private members like private method, variable of parent class in child class.
- Private members are accessible only inside same class.

18 : What is an interface in java?

- An interface is a blue print of a class which can hold abstract methods (Methods without implementation) only.
- It creates Rules To Follow structure for class where It Is Implemented. We can achieve 100% abstraction using interface in java.

19 : Can we access protected method of parent class in sub class?

- Yes.. We can access protected members of parent class in all it's sub classes and classes within the same package.

20 : What is an array in java?

- An array is container object in java which can hold fixed number of values of same type.

21 : Explain System.out.println();

- System : is a final class in java.lang package.
- out : is a static member of system class. It is an instance of java.io.PrintStream. This stream is already open and ready to accept output data.
- println : is a method of java.io.PrintStream .It is an overloaded method.

22 : Write program to print fibonacci series 0, 1, 1, 2, 3, 5, 8, 13, 21,...

```
1 public class fibonacci {  
2  
3 public static void main(String[] args) {  
4  
5 int f1 = 0;
```

```
6
7  int f2 = 1;
8
9  int sum = 0;
10
11 for(int i=0; i<=21;){
12
13 System.out.println(f1);
14
15 sum = i+f2;
16
17 f2=i;
18
19 f1=sum;
20
21 i=f1;
22
23 }
24
25 }
26
27 }
```

23 : Write program to get result of 52+42-32+22-12

```
1  public class SquareSum {
2
3  public static void main(String[] args) {
4
5  int sum = 0;
6
7  for (int i=5;i>=1;i--){
8
9  if(i%2!=0){
10
11 if(sum<(i*i)){
12
13 sum = (i*i)-sum;
14
15 }else{
16
17 sum = sum-(i*i);
18
19 }
20
21 }else{
22
23 sum = sum+(i*i);
24
25 }
26
27 }
28
29 System.out.println(sum);
30
31 }
32
33 }
```

24 : What is return type of testing @DataProvider annotation method?

- It will return double object array "Object[][]".

25 : int x=10 and y=20. Swap both variable values without using any temp variable.

```
1 public class swapNumbers {
2
3 public static void main(String[] args) {
4
5 int x = 10;
6
7 int y = 20;
8
9 System.out.println("Before swapping x = " + x + " and y = " + y);
10
11 x = x + y;
12
13 y = x - y;
14
15 x = x - y;
16
17 System.out.println("After swapping x = " + x + " and y = " + y);
18
19 }
20
21 }
```

26 : What is local variable in java?

- Local variable is declared inside method or constructor and it is limited for that method or constructor only.

Local Variable Example :

```
1 public class JavaVariable {
2
3 public void Calc() {
4
5 // Local Variables.
6
7 int sum;
8
9 int sum1= 5;
10
11 int sum2= 7;
```

```
12
13 sum = sum1+ sum2;
14
15 System.out.println("Sum is : " + sum);
16
17 }
18
19 public static void main(String args[]) {
20
21     JavaVariable jv = new JavaVariable();
22
23     jv.Calc();
24
25 }
26
27 }
```

27 : What is a Instance Variable in java?

- **Instance Variable** is declared parallel to method or constructor in class. It is visible for all methods and constructors of that class.

Instance Variable Example :

```
1 public class JavaVariable {
2
3     //Instance Variable.
4
5     int sum;
6
7     public void Calc() {
8
9         // Local Variables.
10
11         int sum1 = 5;
12
13         int sum2 = 7;
14
15         sum = sum1 + sum2;
16
17         System.out.println("Sum is : " + sum);
18
19     }
20
21     public static void main(String args[]) {
22
23         JavaVariable jv = new JavaVariable();
24
25         jv.Calc();
26
27     }
```

28 : What is a Class Variable in java?

- **Class variable is declared with static keyword in class parallel to methods and constructor.**
- **Class variable is initialized only once at the start of execution and destroyed on end of program.**
- **Class variable is also known as static variable.**

Class Variable Example :

```
1  public class JavaVariable {
2
3  //Class Variable
4
5  static String sumType = "Basket";
6
7  //Instance Variable.
8
9  int sum;
10
11 public void Calc() {
12
13 // Local Variables.
14
15 int item1 = 5;
16
17 int item2 = 7;
18
19 sum = item1 + item2;
20
21 System.out.println(sumType + " Sum is : " + sum);
22
23 }
24
25 public static void main(String args[]) {
26
27 JavaVariable jv = new JavaVariable();
28
29 jv.Calc();
30
31 }
```

29 : What is the difference between instance variable and class variable?

Instance Variable :

- **It is unique to each instance of the class.**
- **Declared without static modifier.**
- **Memory allocation, loading and initialization is done at run time.**

Static Variable :

- It is shared by all instances of the class.
- Declared with static modifier.
- Memory allocation is done at compile time, loaded at load time and they are initialized at class initialization time.

30 : What is access modifier in java?

- Access modifiers allows us to set access levels for variables, methods, classes and constructors in java.
- We can control access levels using access modifiers in java.

31 : What is default value of local variable.

- There is not any default value of local variable. You must have to initialize it.

32 : Java support constructor inheritance?

- No, Constructor inheritance is not supported in java.

33 : Which is super class of all other classes in java?

- java.lang.Object is super class of all other classes in java.

34 : What is Encapsulation?

- Encapsulation is process of packing code and data together In to a single Unit.

35 : Write a program to reverse a string without using reverse function.

```
1 package JAVAExamples;
2
3 public class StringReverse {
4
5     public static void main(String[] args) {
6
7         //String to reverse.
8
9         String str = "This Is String.";
10
11        String revstring = "";
12
13        for (int i = str.length() - 1; i >= 0; --i) {
14
15            //Start getting characters from end of the string.
16
17            revstring += str.charAt(i);
18
19        }
20
21        System.out.println(revstring);
22
23    }
24
25 }
```

Output : .

```
1 gnirtS sI sihT
```

36 : What is the difference between the Constructor and Method?

Constructor :

- Name of the constructor must be same as class name.
- Constructor must not have any return type.
- It is used to initialize the state of an object.
- It is not possible to call constructor directly. Constructors called implicitly when the new keyword creates an object.

Method :

- Method name can be any.
- Method must have return type.
- It is used to expose behavior of an object.
- Methods can be called directly.

- **Read more on Constructor and Method.**

37 : What are the different types of types of constructors in java?

Default Constructor : Constructor without parameter is called default constructor.

```
1 public class City {
2
3 //Default Constructor
4
5 City()
6
7 {
8
9 System.out.println("City is created");
10
11 }
12
13 public static void main(String args[]){
14
15 City c=new City();
16
17 }
18
19 }
```

Parameterized constructor : Constructor with parameter is called Parameterized constructor.

```
1 public class City {
2
3 int id;
4
5 String name;
6
7 // parameterized Constructor
8
9 City(int i, String n) {
10
11 id = i;
12
13 name = n;
14
15 }
16
17 void display() {
18
19 System.out.println(id + " " + name);
20
21 }
22
23 public static void main(String args[]) {
24
25 City c1 = new City(1, "New York");
26
27 City c2 = new City(2, "London");
28
29 c1.display();
30 }
```

```
30
31 c2.display();
32
33 }
34
35 }
```

Output :

```
1 1 New York
2
3 2 London
```

38 : Write a program for Fibonacci series in Java ?

```
1 public class FibonacciSeries {
2
3 public static void main(String args[]) {
4
5 int x1 = 0, x2 = 1, x3, i, cnt = 15;
6
7 // To print 0 and 1
8
9 System.out.print(x1 + " " + x2);
10
11 // loop starts from 2 as 0 and 1 are already printed.
12
13 for (i = 2; i < cnt; ++i) {
14
15 x3 = x1 + x2;
16
17 System.out.print(" " + x3);
18
19 x1 = x2;
20
21 x2 = x3;
22
23 }
24
25 }
26
27 }
```

39 : Write a program to print below given pattern.

```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```

Program to print above pattern is as bellow.

```
1 public class Pattern {
2
3 public static void main(String[] args) {
4
5 for (int a = 1; a <= 5; a++) {
6
7 for (int x = 1; x <= a; x++) {
8
9 System.out.print(x+" ");
10
11 }
12
13 // To print new line.
14
15 System.out.println();
16
17 }
18
19 }
20
21 }
```

40 : What is the difference between “this” and “super” keywords in Java?

- “this” keyword is used to store current object reference while “super” keyword is used to store super class object in sub class..
- “this” is used to access methods of the current class while “super” is used to access methods of the base class.
- this() used to call constructors in the same class whereas super() is used to call super class constructor.

41 : In java, What is return type of main method?

- Main method doesn't have any return type. It is void.

42 : Can We Overload main method in java?

- Yes, Java class can have any number of main methods so it is possible to overload main method.
- But when you run program, It will not execute overloaded main method.
- Always It will execute only public static void main(String[] args) method.

43 : Can we declare class as protected?

- No, You can not declare class as protected.

44 : Write a program to remove given character from string.

```
1 public class RemoveChar {
2
3     public static String removeChar(String str, char c) {
4
5         if (str == null)
6
7             return null;
8
9         return str.replaceAll(Character.toString(c), "");
10
11     }
12
13     public static void main(String[] args) {
14
15         System.out.println(removeChar("chicago", 'c'));
16
17     }
18
19 }
```

45 : How to convert string from upper to lower and lower to upper case?

- You can use toUpperCase and toLowerCase methods to convert string from lower to upper and upper to lower case.

46 : What is Polymorphism?

- Polymorphism is ability using which we can create reference variables or methods which behaves differently in different programmatic context.
- Best example of polymorphism is human.
- We behaves differently with different people in different environment.
- Our behavior will be different when we meet to boss and meet to friend.

47 : What is the advantages of Polymorphism?

- Main advantage of polymorphism is code reusability.
- You can dynamically supply different implementations through polymorphism.
- So it will reduce your work volume in terms of handling and distinguishing various objects.

48 : What is a package?

- A package is a namespace which allows developer to organizes a group of related classes and interfaces.
- Conceptually it is just like folder which contains different types of files.
- It is easy to keep things organised by keeping related classes and interfaces into packages.

49 : What is string in java?

- In Java programming, String is object which is prepared by sequence of characters. `java.lang` package has String class to create and manipulate strings.

50 : What is StringBuffer in java?

- StringBuffer help us to create mutable(modifiable) string in java.
- That means we can modify the string if we use StringBuffer.