1. Can abstract class have constructors in Java?

Ans: yes

1. Can abstract class implements interface in Java? do they require to implement all methods?

Ans: Yes, No

3)  Can abstract class be final in Java?

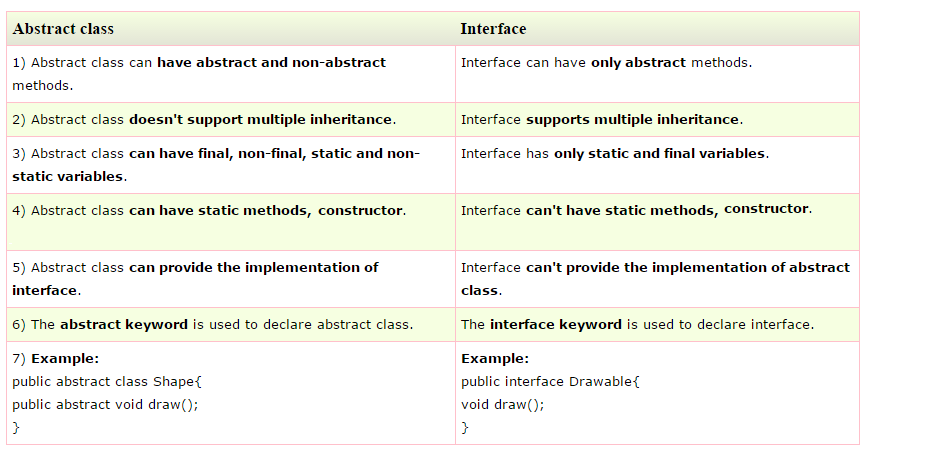
Ans: No  
4)  Can abstract class have static methods in Java?

Ans: Yes  
5)  Can you create instance of abstract class?

Ans: No  
6)  Is it necessary for abstract class to have abstract method?

Ans: Yes  
7)  Difference between abstract class and interface in Java?

Ans:



8)  When do you favor abstract class over interface?

Ans: Abstract class can provide default behaviour where Interfaces cannot. This makesens when a part of the behaviour will be common accross several subclasses.

9)    What is abstract method in Java?

Ans: **Abstract** classes cannot be instantiated, but they can be subclassed. An **abstract method** is a **method** that is declared without an implementation (without braces, and followed by a semicolon), like this: **abstract** void moveTo(double deltaX, double deltaY);

10) Can abstract class contains main method in Java ?

Ans: yes

11)  what is static block in java?

Ans: If we declare a **Static block in java** class it is executed when class loads. This is initialize with the **static** variables. It is mostly used in JDBC. **Static block in java** is executed every time when a class loads.**Static block in java** initializes when class load into memory , it means when JVM read the byte code.

12)  What is the need of static block?

Ans: **Static block** is used for initializing the **static**variables.This**block** gets executed when the class is loaded in the memory. A class can **have** multiple **Static blocks**, which will execute in the same sequence in which they **have** been written into the program.

13)  Can we overload static methods in java?

Ans: **Static methods** cannot be overridden because they are not dispatched on the object instance at runtime.

14)  Can we call super class static methods from sub class?

Ans: Yes

15)What is the difference between final and static keywords?

Ans:

**Final:**  
In very simple terms it means the values once assigned cannot be modified. In the case of final variables, they should either be assigned at declaration or in the constructor. In the case of final classes, it means that they cannot be subclassed.  
  
**Static:**  
Static basically means that the values would be stored in the class memory. So if a class has a static variable, no matter how many instances of it you create, they all would have the same value for the variable. In other words, they all refer to the same copy. Static variables and methods can be accessed directly through the class. Though accessing them through instances wouldn't break compilation, your IDE will complain asking you to access them in a static way. Static methods can access only static variables in the class.

16) Write a note on covariant return type with example code.

Ans:

Ex:

class A

{

A get(){return this;}

}

class B1 extends A

{

B1 get()

{return this;}

void message(){System.out.println("welcome to covariant return type");}

}

public class Jan3

{

public static void main(String[] args)

{

new B1().get().message();

}

}

17) Write a note on Enum with example code.

Ans: The covariant return type specifies that the return type may vary in the same direction as the subclass.

**Enum in java** is a data type that contains fixed set of constants.

It can be used for days of the week (SUNDAY, MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY and SATURDAY) , directions (NORTH, SOUTH, EAST and WEST) etc. The java enum constants are static and final implicitly. It is available from JDK 1.5.

Java Enums can be thought of as classes that have fixed set of constants.

Ex:

package javaapplication26;

enum days{mon,tue,wed,thus,fri,sat,sun}

public class JavaApplication26 {

public static void main(String[] args)

{

for(days s:days.values()) //The values() method returns an array containing all the values of the enum.

{

System.out.println(" "+s);

}

}

}

18) Write a note on use of super keyword and super() method.

Ans:

1. super can be used to refer immediate parent class instance variable.
2. super can be used to invoke immediate parent class method.
3. super() can be used to invoke immediate parent class constructor.

19)  Write a code to implement abstraction using interface.

Ans: ----student.java----

package pp;

public class Student

{

private String name;

private int rollno;

public void setName(String n)

{

this.name=n;

}

public void setRollno()

{

this.name=name;

}

public String getName()

{

return name;

}

public int getRollno()

{

return rollno;

}

Student(String name,int rollno)

{

this.name=name;

this.rollno=rollno;

}

}

----StudentDao.java----(Interface)

package pp;

import java.util.\*;

public interface StudentDao

{

List<Student> getAllStudents();

Student getStudent(int rollno);

void updateStudent(Student s);

void deleteStudent(Student s);

}

-----StudentDaoimpl.java---

package pp;

import java.util.\*;

public class StudentDaoImpl implements StudentDao

{

List<Student> ss;

public StudentDaoImpl()

{

ss=new ArrayList<Student>();

Student s1=new Student("Robert",0);

Student s2=new Student("priyanka",1);

ss.add(s1);

ss.add(s2);

}

@Override

public List<Student> getAllStudents()

{

return ss;

}

@Override

public Student getStudent(int rollno)

{

return ss.get(rollno);

}

@Override

public void updateStudent(Student s)

{

ss.get(s.getRollno()).setName(s.getName());

System.out.println("Student: roll no"+s.getRollno()+"updated in DB");

}

@Override

public void deleteStudent(Student s)

{

ss.remove(s.getRollno());

System.out.println("Student: roll no"+s.getRollno()+"Deleted in DB");

}

}

----EndUserCall.java---

package pp;

public class EndUserCall

{

public static void main(String ss)

{

StudentDao sdao=new StudentDaoImpl();

for(Student s: sdao.getAllStudents())

{

System.out.println(s.getRollno()+" "+s.getName());

}

Student student=sdao.getAllStudents().get(0);

student.setName("suraj");

sdao.updateStudent(student);

sdao.getStudent(0);

System.out.println(student.getRollno()+" "+student.getName());

}

}

20)Write a Java program to sort a numeric array and a string array.

Ans:

package jan3;

importjava.util.Arrays;

public class Jan3

{

public static void main(String[] args)

{

int[] my\_array1 = {

17, 20, 18, 14, 21,

48, 24, 12, 72, 23,

56, 16, 47, 46};

String[] my\_array2 = {

"Java",

"Python",

"PHP",

"C#",

"C Programming",

"C++"

};

System.out.println("Original numeric array : "+Arrays.toString(my\_array1));

Arrays.sort(my\_array1);

System.out.println("Sorted numeric array : "+Arrays.toString(my\_array1));

System.out.println("Original string array : "+Arrays.toString(my\_array2));

Arrays.sort(my\_array2);

System.out.println("Sorted string array : "+Arrays.toString(my\_array2));

}

}

21)Write a Java program to sum values of an array.

Ans:

package jan3;

public class Jan3

{

public static void main(String[] args)

{

intmy\_array[] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};

int sum = 0;

for (int i : my\_array)

sum += i;

System.out.println("The sum is " + sum);

}

}

22)Write a Java program to remove a specific element from an array.

Ans:

package jan3;

importjava.util.Arrays;

public class Jan3

{

public static void main(String[] args)

{

int[] my\_array = {5, 4, 6, 15, 36, 56, 7, 8, 9, 49};

System.out.println("Original Array : "+Arrays.toString(my\_array));

// Remove the second element (index->1, value->14) of the array

intremoveIndex = 1;

for(int i = removeIndex; i <my\_array.length -1; i++){

my\_array[i] = my\_array[i + 1];

}

// We cannot alter the size of an array , after the removal, the last and second last element in the array will exist twice

System.out.println("After removing the second element: "+Arrays.toString(my\_array));

}

}

23)Write a Java program to reverse an array of integer values.

Ans: package jan3;

importjava.util.Arrays;

public class Jan3

{

public static void main(String[] args)

{

int[] my\_array1 = {

1, 2, 3, 4, 5, 6, 7, 8, 9, 10};

System.out.println("Original array : "+Arrays.toString(my\_array1));

for(int i = 0; i < my\_array1.length / 2; i++)

{

int temp = my\_array1[i];

my\_array1[i] = my\_array1[my\_array1.length - i - 1];

my\_array1[my\_array1.length - i - 1] = temp;

}

System.out.println("Reverse array : "+Arrays.toString(my\_array1));

}

}

24)Write a Java program to find the duplicate values of an array of integer values.

Ans:

package jan3;

importjava.util.Arrays;

public class Jan3

{

public static void main(String[] args)

{

int[] my\_array = {1, 2, 5, 5, 6, 6, 7, 2};

for (int i = 0; i < my\_array.length-1; i++)

{

for (int j = i+1; j <my\_array.length; j++)

{

if ((my\_array[i] == my\_array[j]) && (i != j))

System.out.println("Duplicate Element : "+my\_array[j]);

}

}

}

}