1) Can abstract class have constructors in Java?

Yes,abstract class can have constructors.

2) Can abstract class implements interface in Java? do they require to implement all methods?

Abstract class can implement interface in java,but it is not required to implement all methods.

3) Can abstract class be final in Java?

Abstract class cannot be final in java as abstract class needs to be subclassed.

4) Can abstract class have static methods in Java?

Abstract class can have concrete methods that can be declared as static.

5) Can you create instance of abstract class?

Abstrast method cannot be instantiated.

6) Is it necessary for abstract class to have abstract method?

Abstract class can be created without abstract method.

7) Difference between abstract class and interface in Java?

Abstract is declared using abstract keyword .It can have abstract and concrete methods.It can have final,non final ,static or non static variables.

Interface is delared using interface keyword.It can have only abstract methods.The variables declaared in interface are static and final.

8) When do you favor abstract class over interface?

Abstract class is preferred when there is a need to have same functionality for certain methods in the related child classes.

9) What is abstract method in Java?

Abstract methods are those for which there is no implementation.

10) Can abstract class contains main method in Java ?

Yes it is possible to have main method in abstract class.

11) what is static block in java?

Static block is declared as

static{

//statements

}.This block gets executed first when the class is loaded in the memory.

12) What is the need of static block?

It is mainly used for initialising static variables.This block is executed first.

13) Can we overload static methods in java?

No.

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

yes.

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

Final is used with variables,methods and class.When final used with variable it means value is constant,with method it cannot be overriden and the clas cannot be inherited.Static is used variables ,methods and only nested classes.Static variables and methods are not associated with any instance and it is maintained throughout the application

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

For method overriding the sub class method should have same return type,name and arguments as super class method .But from java 5,the return type of sub class can be any sub type of the super class return type.

package covar;

class Shape

{

Shape show()

{

return this;

}

}

public class Circle extends Shape{

Circle show()

{

return this;

}

void display()

{

System.out.print("display method");

}

public static void main(String[] args) {

new Circle().show().display();

}

}

17) Write a note on Enum with example code.

Enum is a kind of datatype that represents a set of named constants.They can be used to represent days of week or cards etc. Syntax is

Enum enumname{const1,const2,….constn } where enumname is name of enum and const1,const2…constn represents the constant values.For ex.

Enum Day{Sunday,Monday,Tuesday,Wednesday,Thursday,Friday,Saturday}

It is similar to class and can have variables,methodsand constructors.It can be declared inside the class or outside the class.

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

The keyword super is used to access the members of immediate super class from sub class.Syntax is super.variable or super.method().super() is used to refer the super class constructor.This should be the first statement in sub class constructor.

class super1

{

int val=10;

void show()

{

System.out.print("super");

}

}

public class sub1 extends super1{

int val=20;

void show()

{

super.show();

System.out.print("sub"); }

void display()

{

System.out.print("Sub value"+val);

System.out.print("Super value"+super.val);

}

public static void main(String[] args) {

sub1 t=new sub1();

t.show();

t.display();

}

}

19)  Write a code to implement abstraction using interface.

Interface is used to implement abstraction as interface contains methods without implementation.When it is implemented in class we provide implementation for the method.

interface interface1

{

void show();

void message();

}

public class Test implements interface1{

public void show()

{

System.out.print("meth 1");

}

public void message()

{

System.out.print("meth 2");

}

public static void main(String[] args) {

Test t=new Test();

t.show();

t.message();

}

}

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

a)Sort numeric array

import java.util.Scanner;

public class Numsort {

public static void main(String[] args) {

Scanner s=new Scanner(System.in);

System.out.println("Enter array size :");

int n=s.nextInt();

int temp;

int num[]=new int[n];

for(int i=0;i<n;i++){

num[i]=s.nextInt();

}

for(int i=0;i<n-1;i++){

for(int j=i+1;j<n;j++){

if(num[i]>num[j]){

temp=num[i];

num[i]=num[j];

num[j]=temp;

}

}

}

for(int i:num)

System.out.println(i);

}

}

b)Sort String array.

import java.util.Scanner;

public class Strsort {

public static void main(String[] args) {

Scanner s=new Scanner(System.in);

System.out.println("Enter array size :");

int n=s.nextInt();

String names[]=new String[n];

for(int i=0;i<n;i++){

names[i]=s.next();

}

for(int i=0;i<n-1;i++){

for(int j=i+1;j<n;j++){

if(names[i].compareTo(names[j])>0){

String temp=names[i];

names[i]=names[j];

names[j]=temp;

}

}

}

for(String sname:names)

System.out.println(sname);

}

}

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

import java.util.Scanner;

public class Arrsum{

public static void main(String[] args) {

Scanner s=new Scanner(System.in);

System.out.println("Enter array size :");

int n=s.nextInt();

int sum=0;

int num[]=new int[n];

for(int i=0;i<n;i++){

num[i]=s.nextInt();

}

for(int i : num){

sum+=i;

}

System.out.println("Sum of elements of array " +sum);

}

}

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

import java.util.Scanner;

public class Arrremove {

public static void main(String[] args) {

Scanner s=new Scanner(System.in);

System.out.println("Enter array size :");

int n=s.nextInt();

System.out.println("Enter element to remove :");

int m=s.nextInt();

int num[]=new int[n];

for(int i=0;i<n;i++){

num[i]=s.nextInt();

}

for(int i=0;i<n;i++){

if(num[i]==m){

num[i]=num[i+1];

}

}

for(int i=0;i<n-1;i++){

System.out.println(num[i]);

}

}

}

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

import java.util.Scanner;

public class arreverse {

public static void main(String[] args) {

Scanner s=new Scanner(System.in);

System.out.println("Enter array size :");

int n=s.nextInt();

int temp;

int num[]=new int[n];

for(int i=0;i<n;i++){

num[i]=s.nextInt();

}

for(int i=0;i<n/2;i++){

temp=num[i];

num[i]=num[n-1-i];

num[n-1-i]=temp;

}

for(int i=0;i<n;i++){

System.out.println(num[i]);

}

}

}

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

import java.util.Scanner;

public class Arrduplicate {

public static void main(String[] args) {

Scanner s=new Scanner(System.in);

System.out.println("Enter array size :");

int n=s.nextInt();

int temp,flag=0,count=0;

int num[]=new int[n];

for(int i=0;i<n;i++){

num[i]=s.nextInt();

}

for(int i=0;i<n-1;i++){

for(int j=i+1;j<n;j++){

if(num[i]==num[j]){

flag=1;count++;

}

}

}

if(flag==0){

System.out.println("No duplicate values");

}

else{

System.out.println(count+" Duplicate values");

}

}

}