**1.Can abstract class have Constructors in Java?**

Yes,abstract class can have a Constructors in Java.

**2.Can Abstract class implements Interface in java?do they require to implement all methods?**

In java,an abstract class can implement an interface and not provide implementation of all of the

Interfaces method.

**3.Can abstract class be final in java?**

Yes,there may be final methods in abstract class,but any abstract method in the class can’t be final.

**4.can abstract class have static method in java?**

Yes,we have static methods in Abstract class,but not abstract method as static.

**5.can you create instance of abstract class?**

No,we can’t create instance for abstract class.

**6.Is it necessary for abstract class to have abstract method?**

No,It is not necessary for abstract class to have abstract method.

**7.Difference between abstract class and Interface in Java?**

|  |  |
| --- | --- |
| Abstract Class | Interface |
| Can have constructors | Can’t have constructors |
| Can have both abstract and concrete methods | Can have only abstract methods |
| Used extends Keyword | Used Implements keyword |
| Members in abstract class can be any Access specifier | Members of a Java interface are public by default. |

**8.What do you favour abstract class over interface?**

Consider using abstract classes if any of these statements apply to your situation:

->You want to share code among several closely related classes.

->You expect that classes that extend your abstract class have many common methods or fields or require access modifiers other than public (such as protected and private).

->You want to declare non-static or non-final fields. This enables you to define methods that can access and modify the state of the object to which they belong.

Consider using interfaces if any of these statements apply to your situation:

->You expect that unrelated classes would implement your interface. For example, the interfaces Comparable and Cloneable are implemented by many unrelated classes.

->You want to specify the behavior of a particular data type, but not concerned about who implements its behavior.

->You want to take advantage of multiple inheritances.

**9.What is abstract method in java?**

Abstract method is the method that does not have implementation.

**10.can abstract class contains main method in Java?**

Yes,we can have abstract class contains main method.

**11.what is static block in Java?**

It is used to initialize the static data member.

It is executed before main method at the time of classloading

**12.what is the need of static block?**

Static block is used for initializing the static variable.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?**

Yes,we can overload static methods in Java.

**14.can we call super class static methods from subclass?**

Yes,we can call super class static method inside subclass.

**15.what is the difference between final and static keywords?**

Final ->

Variable – can’t be changed

*Method – can’t be Overrided*

Class – can’t be Inherited

Static ->

Variable – refer to the common property of all objects

Method – can be invoked without the need for creating an instance of a class.it can access only static variable.

**16.write a note on covariant return type with example code?**

Covariant return means that when one overrides a method,the return type of the overriding method is allowed to be a subtype of the overridden methods returntype.

Class A{

A get()

{  
return this;

}

}

Class B extends A

{  
B get()

{  
return this;

}

Void msg()

{

System.out.println(“Welcome”);

}

Public static void main(String ab[])

{  
new B().get().msg();

}

}

**17.write a note on Enum with Example code.**

Enum is a datatype that contains fixed set of constants.

Class example

{

Enum season{win,sum,spr,fall}

Public static void main(String ab[])

{

For(season s:season.values())

System.out.println(s);

}  
}

**18.write a note on use of super keyword and super() method?**

Super keyword = used to refer immediate Parent class object.

Super() = used to invoke base class constructors

**19.write a code to implement abstraction using Interface.**

Interface Sample

{

Int a=10;

Void area();

}

Class Test implements Sample

{  
public void area()

{  
System.out.println(a\*a);

}

Public static void main(String ab[])

{

Test t=new Test();

t.area();

}

}

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

import java.util.\*;

class Demo

{

public static void main(String ab[])

{

int a[]={12,34,45,89,22,11};  
 String str[]=new String[]{“aaa”,”zzz”,”tyih”,”bbb”};

Arrays.sort(a);

for(int aa:a)

System.out.println(aa);

Arrays.sort(str);

For(String v:str);

System.out.println(v);

}

}

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

import java.util.\*;

class Demo

{

public static void main(String ab[])

{

int a[]={12,34,45,89,22,11};

int sum=0;

for(int b:a)

sum=sum+b;

System.out.println("Sum of an Array : "+sum);

}

}

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

import java.util.\*;

class Demo

{

public static void main(String ab[])

{

int a[]={12,34,45,89,22,11};

Scanner sc=new Scanner(System.in);

int v=sc.nextInt();

int index=0,c=0;

for(int i=0;i<a.length;i++)

{

if(v==a[i])

{

index=i;

for(int j=index;j<a.length-1;j++)

{

a[j]=a[j+1];

}

c++;

break;

}

}

if(c==0)

System.out.println("Not found");

else

{

for(int k=0;k<a.length-1;k++)

System.out.println(a[k]);

}

}

}

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

import java.util.\*;

class Demo

{

public static void main(String ab[])

{

int a[]={12,34,45,89,22,11};

int b[]=new int[a.length];

int j=0;

for(int i=a.length-1;i>=0;i--)

{

//System.out.println(a[i]);

b[j]=a[i];

j++;

}

for(int c:b)

System.out.println(c);

}

}

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

class FindDuplicate

{

public static void main(String ab[])

{

int a[]={1,2,3,2,4,4,3,5,5};

int c=0;

for(int i=0;i<a.length;i++)

{

c=0;

for(int j=i;j<a.length;j++)

{

if(a[i]==a[j])

c++;

}

if(c>=2)

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

}

}

}