1. Can abstract class have constructors in Java?

Yes. Abstract classes have constructor and constructor always invoked when its subclass is instantiated. When instantiating a class, always constructor of that class is used. Every constructor invokes the constructor of its **super class with an implicit call to**super().

And also constructor are also used to initialize fields of a class. And abstract classes may contain fields and sometimes they need to be **initialized by using constructor.**

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

Yes. Abstract class can implement interface and the subclass of abstract class should provide implementation for all the abstract methods present in the abstract class as well as from the interface. It’s not necessary for the abstract class to implement all the methods of interface but for the subclass its mandatory.

3)  Can abstract class be final in Java?

No. The abstract keyword with a class name means the abstract class cannot be instantiated but it’s must to inherit the abstract class and provide implementation for abstract methods. And final class is which cannot be inherited by any other class. If we declare a class as abstract final then it won’t work because it’s necessary to extend the abstract class but final keyword does not allow it.

4)  Can abstract class have static methods in Java?

No. In abstract class the methods maybe abstract or non-abstract methods. The abstract class should be inherited and the subclass should define the abstract methods. But when ans static method is declared with abstract keyword or with only declaration it will give an error because static methods are specific to the class in which they are defined and cannot be overridden or inherited. But its possible to declare a non-abstract static method in abstract class and call it from subclass using class name.

5)  Can you create instance of abstract class?

No. We cannot create an instance of an abstract class because it does not have a complete implementation. The purpose of an abstract class is to provide only declarations by hiding implementation. And subclasses which extends that abstract class provides the implementation which are instantiated to call the methods.

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

No. It’s not necessary to have abstract methods in abstract class.We can have an abstract class with or without abstract methods. And both have different concepts as declaring a class abstract means that it cannot be instantiated and can only be sub classed. Declaring a method abstract means that method will be defined in the subclass.

7)  Difference between abstract class and interface in Java?

|  |  |  |
| --- | --- | --- |
| SR  No. | Abstract class | Interface |
| 1 | Abstract class can have abstract and non-abstract methods. | Interface can have only abstract methods. |
| 2 | Abstract class doesn't support multiple inheritance. | Interface supports multiple inheritance. |
| 3 | Abstract class can have final, non-final, static and non-static variables. | Interface has only static and final 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. |

8)  When do you favor abstract class over interface?

Abstract classes contain one or more abstract methods. The abstract classes are favourable in following situations:

* When you want to share code among several related classes.
* When the classes that extend abstract class have many common methods or fields or require access modifiers other than public such as protected and private.
* When you want to declare non-static or non-final fields.

9)    What is abstract method in Java?

A method declared with abstract keyword is called abstract method in java. The abstract methods have only declaration in the class and they are defined in the subclasses.

Abstract methods are declared in abstract class and in interfaces.

10) Can abstract class contains main method in Java?

Yes. An abstract class can contain main method but as we can’t create instance of the abstract class so it’s no use of using the main method. By taking main method we can only call non abstract static methods of the same abstract class.

11)  What is static block in java?

A block declared with static keyword is called as Static block. 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.

The static block is executed only once as it has only single copy in memory. Static block gets executed when object of a class is created or any static member of class is accessed.

12)  What is the need of static block?

The static blocks are needed for the initialization of static variables. And Also for the execution of the code which always need to be executed first. For example the connection purpose we can use static block.

13)  Can we overload static methods in java?

Yes, we can overload static method in Java. But to overload [static method](http://javarevisited.blogspot.sg/2011/11/static-keyword-method-variable-java.html) you need to provide another static method with same name but different method signature. Because static overloaded method are resolved using [static binding](http://javarevisited.blogspot.com/2012/03/what-is-static-and-dynamic-binding-in.html) during compile time.

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

Yes. We can call the static methods of super from subclass by calling them using the class name. Because we cant call static methods by creating instance of class as its having only single copy in memory.

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

The difference between final and static keyword is:

If we declare a variable as final then we can’t change its value and is as static then value can be changed remain consistent till program flow.

If a method is declared as final then it cannot be overridden and is declared as static then no need to create instace to call that method.

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

By using covariant return type we can change the return type of methods in subclasses.

The return type may vary according to the subclass return types.

e.g.

Class A

{

A get()

{

return this;

}

}

class B1 extends A

{

B1 get()

{

Return this;

}

void message()

{

System.out.println("Welcome to NIIT");

}

public static void main(String args[])

{

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

}

}

Output : Welcome to NIIT.

In above program the return type of get method is covariant return type which is changing in subclass.

17) Write a note on Enum with example code.

Enum in java is a data type that contains fixed set of constants.It can be used for days of the week, directions etc. The java enum constants are static and final implicitly.

class EnumDemo

{

public enum Directions { East,West,South,North }

public static void main(String[] args)

{

for (Directions d : Directions.values())

System.out.println(d);

}

}

Output:

East

West

South

North

In above code there is one enum directions which contains the direction constants. And it displays all the directions bsy using for each loop.

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

Super keyword:

It is used to access the members of immediate super class.

By using we can access varibles as super.variable\_name which access only immediate superclass members. And super keyword is used to call the methods of superclass.

Super() statement:

It is used to call the superclass constructor and its given in constructor body only.

It should be the first statement of constructor.

Multiple super() statements are not allowed.

19)  Write a code to implement abstraction using interface.

// Code to implement absatraction using interface

interface SquareInf

{

Final int num=5;

void DisplaySquare();

}

Class AbstractionDemo implements SquareInf

{

Void DislpaySquare()

{  
 System.out.println(“Sqaure of num ”+num+” is :”+(num\*num));

}

Public static void main(String[] args)

{  
 AbstractionDemo a1=new AbstractionDemo();

a1.DisplaySqaure();

}

}

Output: Sqaure of 5 is : 25

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

import java.util.Arrays;

public class SortArray {

public static void main(String[] args){

int[] arr = { 12,324,23,40,34};

String[] strArr = { “C”,”C++”,”JAVA”,”VB”,”SQL”};

System.out.println("Original numeric array : ");

for(int i=0; i < arr.length-1; i++){

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

}

Arrays.sort(arr);

System.out.println("After sorting numeric array : ");

for(int i=0; i < arr.length-1; i++){

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

}

System.out.println("Original string array : ");

for(int i=0; i < strArr.length-1; i++){

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

}

Arrays.sort(strArr);

System.out.println("After sorting string array : ");

for(int i=0; i < strArr.length-1; i++){

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

}

}

}

Output :

Original numeric array :

12

324

23

40

34

After sorting numeric array :

12

23

34

40

324

Original String array :

C

C++

JAVA

VB

SQL

After sorting string array :

C

C++

JAVA

SQL

VB

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

Import java.util.Scanner;

Class Sum{

Public static void main(String[] args){  
 int arr=new int[5];

Scanner sc =new Scannr(System.in);

Int sum=0;

System.out.println(“Enter arr values : ”);

For(int i=0;i<arr.length-1;i++){  
 arr[i]=sc.nextInt();

}

For(int i: arr){

Sum=sum+I;

}

System.out.println(“Sum of arr values :”+sum);

}

}

Output :

Enter arr values:

10

20

30

40

50

Sum of arr values : 150

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

import java.util.Arrays;

import java.util.Scanner;

public class RemoveElement {

public static void main(String[] args) {

int[] arr1 = {10,12,14,16,35,45};

Scanner sc =new Scannr(System.in);

System.out.println("Original array : ");

for(int i=0; i < arr1.length-1; i++){

System.out.println(arr1[i]);

}

System.out.println(“Enter index to remove element :”);

int index = sc.nextInt();

for(int i = index; i < arr1.length -1; i++){

arr1[i] = arr1[i + 1];

}

System.out.println("After removing element array : ");

for(int i=0; i < arr1.length-1; i++){

System.out.println(arr1[i]);

}

}

}

Output :

Original array:

10

12

14

16

35

45

Enter inex to remove element :

1

After removing element array :

10 14 16 35 45

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

import java.util.Scanner;

public class ReverseArray

{

public static void main(String args[])

{

int size, i, j, temp;

int arr[] = new int[50];

Scanner sc = new Scanner(System.in);

System.out.print("Enter array Size : ");

size = sc.nextInt();

System.out.print("Enter the array Elements : ");

for(i=0; i<size; i++)

{

arr[i] = sc.nextInt();

}

j = i - 1;

i = 0;

while(i<j)

{

temp = arr[i];

arr[i] = arr[j];

arr[j] = temp;

i++;

j--;

}

System.out.print("After reversing the array : ");

for(i=0; i<size; i++)

{

System.out.print(arr[i]+ " ");

}

}

}

Output :

Enter array size :

5

Enter the array elements :

1 2 3 4 5

After reversing the array :

5 4 3 2 1

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

import java.util.Arrays;

public class DuplicateValues {

public static void main(String[] args)

{

int arr=new int[5];

int[] dup;

Scanner sc =new Scannr(System.in);

System.out.println(“Enter the array :”);

for(int i=0;i<arr.length-1;i++)

{  
 arr[i]=sc.nextInt();

}

Int k=0;

for (int i = 0; i < arr.length-1; i++)

{

For (int j = i+1; j < arr.length; j++)

{

if ((arr[i] == arr[j]) && (i != j))

{

dup[k]=arr[j]);

k++;

}

}

}

System.out.println("Duplicate elements : “);

for(int i=0;i<dup.length-1;i++)

{

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

}

}

}

Output:

Enter the array:

10 20 10 30 20

Duplicate Elements:

10

20