1) Can abstract class have constructors in Java?

Ans: Yes, It can be used to initialize final variables.

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

Ans: Yes abstract class can implement interface. No, we need not implement all methods because class is abstract.

3) Can abstract class be final in Java?

Ans: No. Abstract class needs to be extends and its methods must be overridden to use whereas Final class cannot be extended.

4) Can abstract class have static methods in Java?

Ans: Yes it can have static methods, abstract methods in an abstract class cannot be static.

5) Can you create instance of abstract class?

Ans: No, because its definition is not complete.

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

Ans: No, abstract class support 0 to 100 percent level of abstract method.

7) Difference between abstract class and interface in Java?

Ans:

|  |  |
| --- | --- |
| **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?

Ans:

1. You want to share code among several closely related classes.
2. You expect that classes that extend your abstract class have many common methods or fields or require access modifiers other than public.
3. 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.

9) What is abstract method in Java?

Ans:

1. Method with only declaration and no definition in its own class.
2. These methods must be defined in abstract class.

10) Can abstract class contains main method in Java ?

Ans: Yes, Abstract class can contain main method in Java and as it is static you don’t need object to call it.

11) what is static block in java?

Ans: Static block is a block of statements which is executed when class is loaded into memory.

12) What is the need of static block?

Ans: Static block is used to initialize static variables.

13) Can we overload static methods in java?

Ans: Static methods can be overloaded in java and can be executed using class name.

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

Ans: Yes we can call superclass static methods in sub class.

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

Ans: Static keyword creates single copy of variable which associated with class and not object whereas Final keyword will not allow variable change its value once initialized.

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

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

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 static void main(String args[]){

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

}

}

17) Write a note on Enum with example code.

Ans:

1. enum in java is a data type that contains fixed set of constants.
2. enum improves type safety
3. enum can be easily used in switch
4. enum can be traversed
5. enum can have fields, constructors and methods
6. enum may implement many interfaces but cannot extend any class because it internally extends Enum class

Example:

class EnumExample1{

public enum Season { WINTER, SPRING, SUMMER, FALL }

public static void main(String[] args) {

for (Season s : Season.values())

System.out.println(s);

}

}

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

Ans:

1. super keyword is used to refer immediate parent class instance variables.
2. super keyword can also be used to invoke immediate parent class method.
3. super() method is used to execute immediate patent class constructor.

19) Write a code to implement abstraction using interface.

Ans:

interface Bank{

float rateOfInterest();

}

class SBI implements Bank{

public float rateOfInterest(){return 9.15f;}

}

class PNB implements Bank{

public float rateOfInterest(){return 9.7f;}

}

class TestInterface2{

public static void main(String[] args){

Bank b=new SBI();

System.out.println("ROI: "+b.rateOfInterest());

}

}

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

Ans:

**Sorting Numeric Array**

public static void main(String[] args) {

int[] arr = new int[] { 6, 8, 7, 4, 312, 78, 54, 9, 12, 100, 89, 74 };

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

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

int tmp = 0;

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

tmp = arr[i];

arr[i] = arr[j];

arr[j] = tmp;

}

}

}

}

**Sorting String Array**

import java.util.Iterator;

import java.util.SortedSet;

import java.util.TreeSet;

public class SortingDemo

{

public static void main(String[] args) {

SortedSet<String> set = new TreeSet<String>();

String[] s= {"this", "will", "be", "sorted", "without", "any", "sort()", "function", "or", "comparator"};

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

{

set.add(s[i]);

}

Iterator<String> it = set.iterator();

while (it.hasNext()) {

Object element = it.next();

System.out.println(element.toString());

}

}

}

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

Ans:

class Example{

public static void main(String[] args) {

int[] arr = new int[] { 6, 8, 7, 4, 312, 78, 54, 9, 12, 100, 89, 74 };

int sum=0;

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

sum=sum+arr[i]; }

System.out.println(“Sum of Elements of array is ”+sum);

}

}

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

Ans:

Import java.util.\*;

class DeleteElement

{

public static void main(String[] args)

{

int[] arr = new int[] { 6, 8, 7, 4, 312, 78, 54, 9, 12, 100, 89, 74 };

}

}

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

Ans

import java.io.\*;

class ReverseArray {

static void rvereseArray(int arr[], int start, int end)

{

int temp;

if (start >= end)

return;

temp = arr[start];

arr[start] = arr[end];

arr[end] = temp;

rvereseArray(arr, start+1, end-1);

}

static void printArray(int arr[], int size)

{

int i;

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

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

System.out.println("");

}

public static void main (String[] args) {

int arr[] = {1, 2, 3, 4, 5, 6};

printArray(arr, 6);

rvereseArray(arr, 0, 5);

System.out.println("Reversed array is ");

printArray(arr, 6);

}

}

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

Ans:

class RepeatElement

{

    void printRepeating(int arr[], int size)

    {

        int i, j;

        System.out.println("Repeated Elements are :");

        for (i = 0; i < size; i++)

        {

            for (j = i + 1; j < size; j++)

            {

                if (arr[i] == arr[j])

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

            }

        }

    }

    public static void main(String[] args)

    {

        RepeatElement repeat = new RepeatElement();

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

        int arr\_size = arr.length;

        repeat.printRepeating(arr, arr\_size);

    }

}