**TASK034**

**—------------------------------------------**

class Main {

public static void add(int x, int y)

{

System.out.println(x+y);

}

public static void add(int x, int y, int z)

{

System.out.println(x+y+z);

}

public static void main(String[] args)

{

add(10,20);

add(20,20,30);

}

}

Or

**public** **class** TASK034

{

**void** add(**int** x, **int** y)

{

System.***out***.println(x+y);

}

**void** add(**int** x, **int** y, **int** z)

{

System.***out***.println(x+y+z);

}

**public** **static** **void** main(String[] args)

{

TASK034 T = **new** TASK034();

T.add(10,20);

T.add(20,30,24);

}

}

**Output**

**30**

**74**

**TASK035**

**—-------------------------------------------**

**public class TASK035**

**{**

**void add(char x, char y)**

**{**

**System.*out*.println(x+y);**

**}**

**void add(int x, int y)**

**{**

**System.*out*.println(x+y);**

**}**

**public static void main(String[] args)**

**{**

**TASK035 T = new TASK035();**

**T.add('h','a');**

**T.add(20,30);**

**}**

**}**

**Output**

**201**

**50**

**TASK036**

**—--------------------------------------**

**public class TASK035**

**{**

**void add(char x, char y)**

**{**

**System.*out*.println(x+y);**

**}**

**void add(int x, int y)**

**{**

**System.*out*.println(x+y);**

**}**

**public static void main(String[] args)**

**{**

**TASK035 T = new TASK035();**

**T.add('h','a');**

**T.add(20,30);**

**}**

**}**

**Output**

30.45

50.45

**TASK037**

**—----------------------------------**

**class Employee**

**{**

**private int pwd;**

**protected int salary;**

**public int empid;**

**Employee()**

**{**

**System.*out*.println("Employee constructor");**

**}**

**}**

**class Hr extends Employee**

**{**

**void details(){**

**//super.pwd=1234;**

**super.salary=50000;**

**super.empid=10001;**

**//System.out.println(pwd);**

**System.*out*.println(salary);**

**System.*out*.println(empid);**

**}**

**}**

**public class TASK037**

**{**

**public static void main(String[] args)**

**{**

**Hr h = new Hr();**

**h.details();**

**}**

**}**

**Output**

Employee constructor

50000

10001

**TASK038 AND TASK039**

**—-----------------------------------------------------------------**

**public class Main {**

**public static void main(String [] args) {**

**Employee1 e = new company("George W.", "Houston, TX", 43);**

**/\* Following is not allowed and would raise error \*/**

**System.out.println("\n Call mailCheck using Employee reference--");**

**e.mailCheck();**

**}**

**}**

**abstract class Employee1 {**

**private String name;**

**private String address;**

**private int number;**

**public Employee1(String name, String address, int number) {**

**System.out.println("Constructing an Employee");**

**this.name = name;**

**this.address = address;**

**this.number = number;**

**}**

**public double computePay() {**

**System.out.println("Inside Employee computePay");**

**return 0.0;**

**}**

**public void mailCheck() {**

**System.out.println("Mailing a check to " + this.name + " " + this.address);**

**}**

**public String toString() {**

**return name + " " + address + " " + number;**

**}**

**public String getName() {**

**return name;**

**}**

**public String getAddress() {**

**return address;**

**}**

**public void setAddress(String newAddress) {**

**address = newAddress;**

**}**

**public int getNumber() {**

**return number;**

**}**

**}**

**class company extends Employee1{**

**company(String name, String address, int number)**

**{**

**super(name, address, number);**

**}**

**}**

**Output**

Constructing an Employee

Call mailCheck using Employee reference--

Mailing a check to George W. Houston, TX

**TASK040**

**—-------------------------------------**

**abstract class Gadgets {**

**abstract void turnOn();**

**abstract void turnOff();**

**}**

**// Concrete class implementing the abstract methods**

**class TVRemote extends Gadgets {**

**@Override**

**void turnOn() {**

**System.*out*.println("TV is turned ON.");**

**}**

**@Override**

**void turnOff() {**

**System.*out*.println("TV is turned OFF.");**

**}**

**}**

**// Main class to demonstrate abstraction**

**public class TASK040{**

**public static void main(String[ ] args) {**

**Gadgets remote = new TVRemote();**

**remote.turnOn();**

**remote.turnOff();**

**}**

**}**

**Output**

TV is turned ON.

TV is turned OFF.

**TASK041**

**—--------------------------------**

**import java.io.\*;**

**// Interface Declared**

**//Driver Code Ends**

**interface testInterface {**

**// public, static and final**

**final int *tax* = 10;**

**// public and abstract**

**void display();**

**}**

**// Class implementing interface**

**class TestClass implements testInterface {**

**// Implementing the capabilities of**

**// Interface**

**public void display(){**

**System.*out*.println("Myclass");**

**}**

**}**

**class TASK041**

**//Driver Code Starts**

**{**

**public static void main(String[] args)**

**{**

**TestClass t = new TestClass();**

**t.display();**

**System.*out*.println(t.*tax*);**

**}**

**}**

**Output**

Myclass

10

**What is the difference between FINAL or Constant ?**

Java uses final to make variables, methods, or classes unchangeable.

It can be used with:

* **Variables** ➝ value cannot change
* **Methods** ➝ cannot be overridden
* **Classes** ➝ cannot be extended (inherited)

### **Java does not have a const keyword like in C or C++.**

* The const keyword is **reserved** in Java but **not used**.
* All constants in Java are created using final.

Task 021: Home Task

**import java.util.Arrays;**

**public class TASK021**

**{**

**public static void main(String[] args)**

**{**

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

**int n=arr.length;**

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

**{**

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

**}**

**}**

**}**

**Output:**

1

2

3

4

5

*Task 022 - home task*

**class TASK022 {**

**public static void main(String[] args)**

**{**

**int[] arr;**

**arr = new int[5];**

**arr[0] = 10;**

**arr[1] = 20;**

**arr[2] = 30;**

**arr[3] = 40;**

**arr[4] = 50;**

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

**{**

**System.*out*.println("Element at index " + i + " : " + arr[i]);**

**}**

**}**

**}**

**Ouput**

Element at index 0 : 10

Element at index 1 : 20

Element at index 2 : 30

Element at index 3 : 40

Element at index 4 : 50

*Task 023 - home task*

***class*** *Student {*

***public******int*** *roll\_no;*

***public*** *String name;*

*Student(****int*** *Roll\_no, String Name){*

***this****.roll\_no = Roll\_no;*

***this****.name = Name;*

*}*

*}*

***public******class*** *TASK023*

*{*

***public******static******void*** *main( String[] args)*

*{*

*Student[] arr;*

*​arr =* ***new*** *Student[5];*

*arr[0] =* ***new*** *Student(1, "aman");*

*arr[1] =* ***new*** *Student(2, "vaibhav");*

*arr[2] =* ***new*** *Student(3, "shikar");*

*arr[3] =* ***new*** *Student(4, "dharmesh");*

*arr[4] =* ***new*** *Student(5, "mohit");*

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

*{*

*System.out.println("Element at " + i + " " + arr[i].roll\_no + " " + arr[i].name+" ");*

*}*

*}*

*}*

*Output*

Element at 0 : { 1 aman }

Element at 1 : { 2 vaibhav }

Element at 2 : { 3 shikar }

Element at 3 : { 4 dharmesh }

Element at 4 : { 5 mohit }

*TASK024 - Home Task*

***class*** *Student{*

***public*** *String name;*

*Student(String name)*

*{*

***this****.name = name;*

*}*

*@Override*

***public*** *String toString(){*

***return*** *name;*

*}*

*}*

***public******class*** *TASK024*

*{*

***public******static******void*** *main(String[] args)*

*{*

*Student[] myStudents =* ***new*** *Student[]{* ***new*** *Student("Dharma"),****new*** *Student("sanvi"),****new*** *Student("Rupa"),****new*** *Student("Ajay")};*

***for****(Student m:myStudents)*

*{*

*System.****out****.println(m);*

*}*

*}*

*}*

**Output**

Dharma

Sanvi

Rupa

Ajay

TASK025 - HomeTask

**import** java.util.Arrays;

**class** TASK025

{

**public** **static** **void** main(String[] args)

{

**int**[][] arr = **new** **int**[3][3];

System.***out***.println("Rows : " + arr.length);

System.***out***.println("Columns : " + arr[0].length);

}

}

**Output**

Rows : 3

Columns : 3

**TASK026 - HomeTask**

**public class TASK026**

**{**

**public static void main(String[] args)**

**{**

**int arr[][]= {{1,2,3},{4,5,6},{7,8,9}};**

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

**{**

**for (int j=0;j<arr.length;j++)**

**{**

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

**}**

**System.*out*.println();**

**}**

**}**

**}**

**Output**

**1 2 3**

**4 5 6**

**7 8 9**

**TASK027 - HomeTask**

**public class TASK027**

**{**

**public static int sum (int[] arr)**

**{**

**int s=0;**

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

**{**

**s+=arr[i];**

**}**

**System.*out*.println("Sum of Array is : " + s);**

**return s;**

**}**

**public static void main(String[] args)**

**{**

**int[] arr = {9,6,2,1};**

***sum*(arr);**

**}**

**}**

**Output**

**Sum of Array is : 18**

**TASK028 - HomeTask**

**public class TASK028 {**

**public static void main(String[] args)**

**{**

**// TODO Auto-generated method stub**

**int[] arr = *ml*();**

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

**{**

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

**}**

**}**

**public static int[] ml()**

**{**

**return new int[] {1,2,3};**

**}**

**}**

**Output**

**1 2 3**

**TASK029- HomeTask**

**public class TASK029 {**

**public static void main(String[] args)**

**{**

**int intArray[] = {1,2,3,4};**

**int cloneArray[]= intArray.clone();**

**System.*out*.println(intArray==cloneArray);**

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

**{**

**System.*out*.println(cloneArray[i]+" ");**

**}**

**// TODO Auto-generated method stub**

**}**

**}**

**Output**

**false**

**1**

**2**

**3**

**4**

**TASK030 - HomeTask**

**public class TASK030 {**

**public static void main(String[] args)**

**{**

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

**int cloneArray[][]= intArray.clone();**

**System.*out*.println(intArray==cloneArray);**

**System.*out*.println(intArray[0]==cloneArray[0]);**

**System.*out*.println(intArray[1]==cloneArray[1]);**

**// TODO Auto-generated method stub**

**}**

**}**

**Output**

**false**

**true**

**true**