Task 031

class Calculation {

   int z;

   public void addition(int x, int y) {

      z = x + y;

      System.out.println("The sum of the given numbers:"+z);

   }

   public void Subtraction(int x, int y) {

      z = x - y;

      System.out.println("The difference between the given numbers:"+z);

   }

}

public class My\_Calculation extends Calculation {

   public void multiplication(int x, int y) {

      z = x \* y;

      System.out.println("The product of the given numbers:"+z);

   }

   public static void main(String args[]) {

      int a = 20, b = 10;

      My\_Calculation demo = new My\_Calculation();

      demo.addition(a, b);

      demo.Subtraction(a, b);

      demo.multiplication(a, b);

   }

}

public class My\_Calculation2 extends Calculation {

   public void multiplication(int x, int y) {

      z = x \* y;

      System.out.println("The product of the given numbers:"+z);

   }

   public static void main(String args[]) {

      int a = 20, b = 10;

      My\_Calculation2 demo = new My\_Calculation2();

      demo.addition(a, b);

      demo.Subtraction(a, b);

      demo.multiplication(a, b);

   }

}

Inheritance provides - reusability

It avoids - duplication

Multi level inheritance

Clac < ========= My\_calculation < ======= calculation

Class calc extends My\_calculation{

}

Code:

package Task30;  
  
public class MainClass {  
 public static void main(String[] args) {  
 AdvancedCalculator myAdvancedCalculator = new AdvancedCalculator();  
 myAdvancedCalculator.doDivision(5,2);  
 myAdvancedCalculator.doAddition(3,19);  
 }  
}

package Task30;  
  
public class BasicCalculator {  
  
 public void doAddition(int num1,int num2)  
 {  
 int sum = num1 + num2;  
 System.*out*.println("Addition is " + sum);  
 }  
  
 public void doSubtraction(int num1,int num2){  
 int sub = num1 - num2;  
 System.*out*.println("Subtraction is " + sub);  
 }  
  
}

package Task30;  
  
public class AdvancedCalculator extends BasicCalculator {  
 public void doMultiplication(int num1,int num2){  
 int mul = num1 \* num2;  
 System.*out*.println("product is " + mul);  
 }  
  
 public void doDivision(int num1,int num2){  
 int div = num1 / num2;  
 System.*out*.println("division is " + div);  
 }  
}

output:

division is 2

Addition is 22

Process finished with exit code 0

Task 033

—-----------------------------------------------------------------

class Customer {

Void purchage\_list{

Int cost = 40;

String items = “Tomatoes”;

}

}

public class Mart extends Customer {

Void billing(){

String items = “onions”;

Int cost = 30;

}

Psvm (String[] args) {

Super.items = “Potatoes”

Super.cost = 50;

       Sout(items);  ⇒

sout(cost);

sout”(%%%%%%%%%%%%%%”);

Sout(super.items);

sout(super.cost);

}

}

==========================================================

class Customer{

int cost = 40;

String items = "Tomatoes";

Customer(){

System.out.println("Constructor called");

}

    void purchage\_list(){

System.out.println("cost of tomatoes in Customer class is "+ cost);

    }

}

public class Task033 extends Customer {

void billing(){

String items = "onions";

int cost = 30;

super.items = "Potatoes";

super.cost = 50;

super.purchage\_list();

       System.out.println(items);

System.out.println(cost);

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

System.out.println(super.items);

System.out.println(super.cost);

// return 0;

}

public static void main(String[] args){

Customer cobj =new Customer();

cobj.purchage\_list();

Task033 tobj = new Task033();

tobj.billing();

    }

}

Code:

package Task33;  
  
public class Customer {  
 String items = "Tomotoes";  
 int cost = 40;  
 void purchaseList(){  
 System.*out*.println("Items are "+items);  
 }  
}

package Task33;  
  
public class Mart extends Customer{  
  
 public void billing(){  
 super.items = "onions";  
 System.*out*.println("Items are "+items);  
 }  
  
 public static void main(String[] args) {  
 Customer myCustomer = new Customer();  
 myCustomer.purchaseList();  
  
 Mart myMart = new Mart();  
 myMart.purchaseList();  
 myMart.billing();  
  
 myCustomer.purchaseList();  
  
 }  
}

output:

Items are Tomotoes

Items are Tomotoes

Items are onions

Items are Tomotoes

Process finished with exit code 0

Task 034

Void add(int x, int y){

Sout —> x and y values

}

Void add(int x, int y, int z){

Sout —-> x, y, z values

}

psvm(){

add(10,20,30);

add(50,100);

}

Code:

public class Task34 {  
 public static void main(String[] args) {  
 *add*(1,2);  
 *add*(9,10,11);  
 }  
  
 public static void add (int a,int b){  
 int sum = a + b;  
 System.*out*.println("Sum of two mumbers is "+sum);  
 }  
  
 public static void add (int a,int b,int c){  
 int sum = a + b + c;  
 System.*out*.println("Sum of three mumbers is "+sum);  
 }  
}

output:

Sum of two mumbers is 3

Sum of three mumbers is 30

Process finished with exit code 0

Task 035

Void add(char x, char y){

Sout —-> x, y values

}

Void add(int x, int y) {

Sout —> x, y values

}

psvm(){

add(‘d’, ‘a’);

add(100, 100);

}

Code:

public class Task35 {  
 public static void main(String[] args) {  
 *add*(1,2);  
 *add*('a','b');  
 }  
  
 public static void add (int a,int b){  
 int sum = a + b;  
 System.*out*.println("Sum of two numbers is "+sum);  
 }  
  
 public static void add (char a,char b){  
 System.*out*.println("Characters are ");  
 System.*out*.println(a);  
 System.*out*.println(b);  
 }  
}

output:

Sum of two numbers is 3

Characters are

a

b

Process finished with exit code 0

Task 036

Void add(int x, float y){

Sout → x, y values

}

Void add(float x, int y){

Sout  → x, y

}

psvm(){

add(10.50f, 60);

add(100, 80.80f)

}

Code:

public class Task36 {  
  
 public static void main(String[] args) {  
 *add*(1,2.3f);  
 *add*(3.2f,5);  
 }  
  
 public static void add (int a,float b){  
 float sum = a + b;  
 System.*out*.println("Sum of two mumbers is "+sum);  
 }  
  
 public static void add (float a,int b){  
 float sum = a + b;  
 System.*out*.println("Sum of two mumbers is "+sum);  
 }  
}

output:

Sum of two mumbers is 3.3

Sum of two mumbers is 8.2

Process finished with exit code 0

Task 038

/\* File name : AbstractDemo.java \*/

Public class AbstractDemo {

   public static void main(String [] args) {

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

      Employee e = new Employee("George W.", "Houston, TX", 43);

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

      e.mailCheck();

   }

}

abstract class Employee {

   private String name;

   private String address;

   private int number;

   public Employee(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;

   }

}

Code:

package Task38\_39;  
  
public class AbstractDemo {  
  
 public static void main(String [] args) {  
 /\* Following is not allowed and would raise error \*/  
 Employee e = new Employee("George W.", "Houston, TX", 43);  
 System.*out*.println("\n Call mailCheck using Employee reference--");  
 e.mailCheck();  
 }  
}

package Task38\_39;  
  
public class Employee {  
 private String name;  
 private String address;  
 private int number;  
  
 public Employee(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;  
 }  
}

output:

Constructing an Employee

Call mailCheck using Employee reference--

Mailing a check to George W. Houston, TX

Process finished with exit code 0

Task 040

// Working of Abstraction in Java

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.");

    }

}

class ACRemote extends Gadgets {

    @Override

    void turnOn() {

        System.out.println("AC is turned ON.");

    }

    @Override

    void turnOff() {

        System.out.println("AC is turned OFF.");

    }

}

// Main class to demonstrate abstraction

public class Main {

    public static void main(String[] args) {

        Gadgets remote = new TVRemote();

        Gadgets remote = new ACRemote();

       remote.turnOn();

        remote.turnOff();

        Gadgets remote = new FanRemote();

        Gadgets remote = new CoolerRemote();

        remote.turnOn();

        remote.turnOff();

    }

}

