Assignment 4

**Question 1:** There are three classes for box, box weight and box shipment. The member variables for box class are height width and depth and there is a method volume which calculates the volume of the box. The box weight class contains an extra variable weight of the box. The box shipment class contains the above variables and also an extra variable the shipment cost. Write a program in JAVA that will calculate the weight and volume of the box using the concept of multi-level inheritance.

**Code:**

import java.util.\*;

class Box

{

double height, width, depth;

Box(double x, double y, double z)

{

height = x; width = y; depth = z;

}

double volume()

{

return height \* width \* depth;

}

}

class BoxWeight extends Box

{

double weight;

BoxWeight(double a, double b, double c, double d)

{

super(a, b, c);

weight = d;

}

}

class BoxShipment extends BoxWeight

{

double cost;

BoxShipment(double a, double b, double c, double d, double e)

{

super(a, b, c, d);

cost = e;

}

void display()

{

System.out.printf("The dimensions of the box are: %f %f %f\n", height, width, depth);

System.out.printf("The volume, weight and cost is %f %f %f\n", volume(), weight, cost);

}

}

class Test

{

public static void main(String args[])

{

Scanner sc = new Scanner(System.in);

double a, b, c, d, e;

System.out.println("Enter the dimensions of the box");

a = sc.nextDouble();

b = sc.nextDouble();

c = sc.nextDouble();

System.out.println("Enter the weight of the box");

d = sc.nextDouble();

System.out.println("Enter the cost of the box");

e = sc.nextDouble();

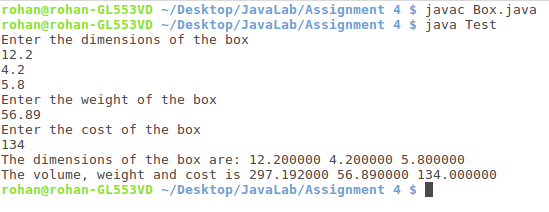
BoxShipment obj = new BoxShipment(a, b, c, d, e);

obj.display();

}

}

**Output:**

**Question 2:** A bank has two kinds of accounts, a Savings account and a Current account. The Savings account provides interest and withdrawal facilities but no cheque book facilities. The Current account provides cheque book facilities but no interest. The Current Account should maintain a minimum balance and if the balance falls below the level, a service charge should be imposed. Create a JAVA class Account that stores customer details. From this derive the classes Savings Account and Current Account more specific to their requirements. Include the necessary functions in order to achieve the following tasks:

1. Accept deposit from a customer and update the balance
2. Display the balance
3. Compute and deposit interest
4. Permit withdrawal and update the balance
5. Check if Current account has minimum balance
6. Impose penalty if required

Write a JAVA program to implement the aforesaid class hierarchy to perform the necessary tasks.

**Code:**

import java.util.\*;

class Account

{

String CustomerName, AccountType; int acc\_no; double balance;

Account(String a, int b, String c, double d)

{

CustomerName = a;

acc\_no = b;

AccountType = c;

balance = d;

}

void deposit(double amt)

{

balance += amt;

}

void display()

{

System.out.println("Customer Name: "+CustomerName);

System.out.println("Accout type: "+AccountType);

System.out.println("Account Number: "+acc\_no);

System.out.println("Balance: "+balance);

}

}

class SavingsAccount extends Account

{

double interest; double rate;

SavingsAccount(String a, int b, String c, double d, double e)

{

super(a, b, c, d);

rate = e;

}

void withdraw(double amt)

{

if( amt > balance )

System.out.println("Insufficient balance!!! Cannot withdraw.");

else

{

balance -= amt;

}

}

void ComputeInterest()

{

interest = rate \* balance / 100;

}

void display()

{

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

super.display();

System.out.println("Interest: "+interest);

System.out.println("Rate: "+rate);

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

}

}

class CurrentAccount extends Account

{

int CheckbookNumber; double ServiceCharge;

CurrentAccount(String a, int b, String c, double d, int ch\_no)

{

super(a, b, c, d);

CheckbookNumber = ch\_no;

ServiceCharge = 0;

}

void CheckMinBalance()

{

if( balance < 500 )

{

System.out.printf("Current balance of Rs. %f is less than 500\n", balance);

System.out.println("Service charge of Rs. 10 has been added");

ServiceCharge += 10;

}

}

void withdraw(double amt)

{

if( amt > balance )

System.out.println("Insufficient balance!!! Cannot withdraw.");

else

{

balance -= amt;

CheckMinBalance();

}

}

void display()

{

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

super.display();

System.out.println("Checkbook number: "+CheckbookNumber);

System.out.println("Service charge: "+ServiceCharge);

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

}

}

class Test

{

public static void main(String args[])

{

SavingsAccount objs = new SavingsAccount("Ram", 1234, "Savings Account", 9000, 0.1);

objs.display();

objs.deposit(100);

objs.withdraw(500);

objs.ComputeInterest();

objs.display();

CurrentAccount objc = new CurrentAccount("Sam", 3412, "Current Account", 500, 12345678);

objc.display();

objc.withdraw(100);

objc.deposit(300);

objc.display();

}

}

**Output:**

