```java
1]/*Develop a Java program to create an abstract class named Shape that
contains two integers and an empty method named printArea( ). Provide
three classes named Rectangle, Triangle and Circle such that each one of
the classes extends the class Shape. Each one of the classes contain only
the method printArea( ) that prints the area of the given shape.*/


abstract class Shape {
    int dim1;
    int dim2;

    Shape(int dim1, int dim2) {
        this.dim1 = dim1;
        this.dim2 = dim2;
    }

    abstract void printArea();
}

class Rectangle extends Shape {
    Rectangle(int length, int breadth) {
        super(length, breadth);
    }

    @Override
    void printArea() {
        int area = dim1 * dim2;
        System.out.println("Area of Rectangle: " + area);
    }
}
```

```
class Triangle extends Shape {
    Triangle(int base, int height) {
        super(base, height);
    }

    @Override
    void printArea() {
        double area = 0.5 * dim1 * dim2;
        System.out.println("Area of Triangle: " + area);
    }
}

class Circle extends Shape {
    Circle(int radius) {
        super(radius, 0);
    }

    @Override
    void printArea() {
        double area = Math.PI * dim1 * dim1;
        System.out.println("Area of Circle: " + area);
    }
}

public class PRG11{
    public static void main(String[] args) {
        Shape rectangle = new Rectangle(10, 5);
        Shape triangle = new Triangle(10, 20);
        Shape circle = new Circle(20);
```

```
        rectangle.printArea();
        triangle.printArea();
        circle.printArea();
    }
}
```

op:

```
Area of Rectangle: 50
Area of Triangle: 100.0
Area of Circle: 1256.6370614359173
```

2]/*Practice programs on Generics on classes and methods separately.*/

```
public class Main {
   public static <T> void printArray(T[] array) {
       for (T element : array) {
           System.out.print(element + " ");
       }
       System.out.println();
   }

   public static void main(String[] args) {
       Integer[] intArray = {1, 2, 3};
       String[] strArray = {"A", "B", "C"};

       System.out.print("Integer array: ");
       printArray(intArray);

       System.out.print("String array: ");
       printArray(strArray);
   }
}
```

Op:

```
Integer array: 1 2 3
String array: A B C
```

```
3]/*Develop a Java program to create a class Bank that maintains two kinds of account
for its
customers, one called savings account and the other current account. The savings
account
provides compound interest and withdrawal facilities but no cheque book facility. The
current
account provides cheque book facility but no interest. Current account holders should
also
maintain a minimum balance and if the balance falls below this level, a service charge
is
imposed.
Create a class Account that stores customer name, account number and type of account.
From
this derive the classes Cur-acct and Sav-acct to make them more specific to their
requirements. Include the necessary methods in order to achieve the following tasks:
a) Accept deposit from customer and update the balance.
b) Display the balance.
c) Compute and deposit interest
d) Permit withdrawal and update the balance
Check for the minimum balance, impose penalty if necessary and update the balance.
*/
abstract class Acc {
   String custName;
   String accNum;
   double ib;

   public Acc(String custName, String accNum, double ib) {
       this.custName = custName;
       this.accNum = accNum;
       this.ib = ib;
   }

   public void deposit(double amount) {
       ib += amount;
       System.out.println("Deposited: " + amount + ". Bal: " + ib);
   }

   public void displayBalance() {
       System.out.println("Bal: " + ib);
   }

   public abstract void withdraw(double amount);
```

```
    public abstract void computeInterest();
}

class SavAcct extends Acc {
    private double interestRate;

    public SavAcct(String custName, String accNum, double ib, double interestRate) {
        super(custName, accNum, ib);
        this.interestRate = interestRate;
    }

    @Override
    public void computeInterest() {
        double interest = ib * interestRate / 100;
        ib += interest;
        System.out.println("Interest added. Bal: " + ib);
    }

    @Override
    public void withdraw(double amount) {
        if (amount <= ib) {
            ib -= amount;
            System.out.println("Withdrawn: " + amount + ". Bal: " + ib);
        } else {
            System.out.println("Insuff. Bal.");
        }
    }
}

class CurAcct extends Acc {
    private static final double MINIMUM_BALANCE = 500.0;
    private static final double SERVICE_CHARGE = 50.0;

    public CurAcct(String custName, String accNum, double ib) {
        super(custName, accNum, ib);
    }

    @Override
    public void computeInterest() {
        System.out.println("No interest on CurAcct.");
    }
```

```
    @Override
    public void withdraw(double amount) {
        if (amount <= ib) {
            ib -= amount;
            System.out.println("Withdrawn: " + amount + ". Bal: " + ib);
            if (ib < MINIMUM_BALANCE) {
                ib -= SERVICE_CHARGE;
                System.out.println("Svc charge applied. Bal: " + ib);
            }
        } else {
            System.out.println("Insuff. Bal.");
        }
    }
}

public class Bank {
    public static void main(String[] args) {
        SavAcct savingsAccount = new SavAcct("Rahul", "S56789", 2000, 4);
        savingsAccount.deposit(500);
        savingsAccount.computeInterest();
        savingsAccount.withdraw(300);
        savingsAccount.displayBalance();

        CurAcct currentAccount = new CurAcct("Tarun", "16102004", 5000);
        currentAccount.deposit(300);
        currentAccount.computeInterest();
        currentAccount.withdraw(400);
        currentAccount.withdraw(200);
        currentAccount.displayBalance();
    }
}
Op:
Deposited: 500.0. Bal: 2500.0
Interest added. Bal: 2600.0
Withdrawn: 300.0. Bal: 2300.0
Bal: 2300.0
Deposited: 300.0. Bal: 5300.0
No interest on CurAcct.
Withdrawn: 400.0. Bal: 4900.0
Withdrawn: 200.0. Bal: 4700.0
Bal: 4700.0
```

[redacted]