

1) CREATING A CLASS PERSON

PROGRAM:

```
import java.util.Scanner;

class person
{
    public String name;
    public int height,weight;

}

class student extends person
{
    int sub1,sub2,sub3,total;
    String study;
    String grade;
    public student()
    {
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter mane of student :");
        name=sc.nextLine();
        System.out.println("Enter class of student");
        study=sc.nextLine();
        System.out.println("Enter sub1 marks :");
        sub1=sc.nextInt();
```

```
System.out.println("Enter sub2 marks :");
sub2=sc.nextInt();
System.out.println("Enter sub2 marks :");
sub3=sc.nextInt();
total=sub1+sub2+sub3;
grade="A";
if(total>90)
{
    grade="A";
}
if(total<90 && total>75)
{
    grade="B";
}
if(total<75 && total>50)
{
    grade="C";
}
if(total<50)
{
    grade="Fail";
}
}
}
```

```
public class test
{
    public static void main(String args[])

    {
        student s=new student();
        System.out.println("\n\n"+"Displaying student information ");
        System.out.println("Student name is :"+s.name);
        System.out.println("Student class is "+s.study);
        System.out.println("Student total marks :"+s.total);
        System.out.println("Student grade :"+s.grade);
    }
}
```

OUTPUT:

```
Enter mane of student :  
SAI  
Enter class of student  
JAVA  
Enter sub1 marks :  
95  
Enter sub2 marks :  
90  
Enter sub2 marks :  
90  
  
Displaying student information  
Student name is :SAI  
Student class is JAVA  
Student total marks :275  
Student grade :A
```

2)CREATING A CLASS SHAPE USING INTERFACE.

PROGRAM:

```
import java.util.Scanner;  
  
interface shape  
{  
    double getarea();  
}  
  
class circle implements shape  
{  
    double radius,pi=3.14;
```

```

public circle()
{
    Scanner sc=new Scanner(System.in);
    System.out.println("Enter the radius of circle");
    radius=sc.nextDouble();
}

@Override
public double getarea()
{
    return pi*radius*radius;
}
}

class triangle implements shape
{
    double base,height;
    public triangle()
    {
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter base of triangle");
        base=sc.nextDouble();
        System.out.println("enter height of triangle");
        height=sc.nextDouble();
    }

    @Override

```

```

    public double getarea()
    {
        return 1/2*base*height;
    }
}

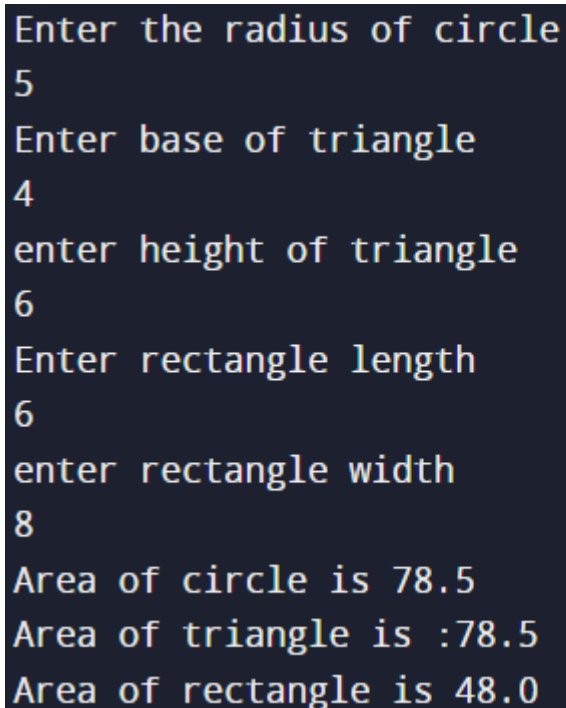
class rectangle implements shape
{
    double width,length;
    public rectangle()
    {
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter rectangle length");
        length=sc.nextDouble();
        System.out.println("enter rectangle width");
        width=sc.nextDouble();
    }
    @Override
    public double getarea()
    {
        return length*width;
    }
}

public class test
{

```

```
public static void main(String args[])
{
    circle c=new circle();
    triangle t=new triangle();
    rectangle r=new rectangle();
    System.out.println("Area of circle is "+c.getarea());
    System.out.println("Area of triangle is :"+c.getarea());
    System.out.println("Area of rectangle is "+r.getarea());
}
}
```

OUTPUT:



```
Enter the radius of circle
5
Enter base of triangle
4
enter height of triangle
6
Enter rectangle length
6
enter rectangle width
8
Area of circle is 78.5
Area of triangle is :78.5
Area of rectangle is 48.0
```

3)CREATING A CLASS NAME DOG AND FIND THE DOG IS BARKING.

PROGRAM:

```

interface Animal
{
    void bark();
}

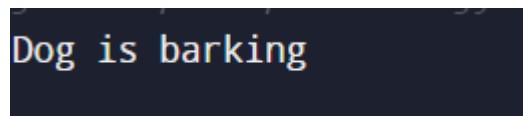
class Dog implements Animal {
    @Override
    public void bark() {
        System.out.println("Dog is barking");
    }
}

public class AnimalExample {
    public static void main(String[] args) {

        Animal dog = new Dog();
        dog.bark();
    }
}

```

OUTPUT:



```

Dog is barking

```

4) write a Java Programing to create a banking system with three classes. Bank account Savings Account, and Current Account the bonk should have al is to of accounts and methals for adding them. Account Should be an interfa -ce with methods to deposit withdraw calculate interest, and view balance, Sovings Account and Current Account should implement the Account interface and have their unique methods.

PROGRAM:


```
import java.util.ArrayList;
import java.util.List;
interface Account {
    void deposit(double amount);
    void withdraw(double amount);
    void calculateInterest();
    double viewBalance();
}
abstract class BankAccount implements Account {
    protected double balance;
    public BankAccount(double initialBalance) {
        this.balance = initialBalance;
    }
    @Override
    public void deposit(double amount) {
        balance += amount;
    }
    @Override
    public void withdraw(double amount) {
        if (amount <= balance) {
            balance -= amount;
        } else {
            System.out.println("Insufficient funds");
        }
    }
}
```

```

    }

    @Override
    public double viewBalance() {
        return balance;
    }

    @Override
    public abstract void calculateInterest();
}

class SavingsAccount extends BankAccount {
    private static final double INTEREST_RATE = 0.05; // 5% annual
    interest rate

    public SavingsAccount(double initialBalance) {
        super(initialBalance);
    }

    @Override
    public void calculateInterest() {
        balance += balance * INTEREST_RATE;
    }
}

class CurrentAccount extends BankAccount {
    private static final double OVERDRAFT_LIMIT = 500.0;

    public CurrentAccount(double initialBalance) {
        super(initialBalance);
    }
}

```

```

    }

    @Override
    public void withdraw(double amount) {
        if (amount <= balance + OVERDRAFT_LIMIT) {
            balance -= amount;
        } else {
            System.out.println("Overdraft limit exceeded");
        }
    }

    @Override
    public void calculateInterest() {
        // Current accounts do not earn interest
    }
}

class Bank {
    private List<Account> accounts = new ArrayList<>();

    public void addAccount(Account account) {
        accounts.add(account);
    }

    public void displayBalances() {
        for (Account account : accounts) {
            System.out.println("Balance: " + account.viewBalance());
        }
    }
}

```

```
    }  
}  
  
public class BankingSystem {  
    public static void main(String[] args) {  
        Bank bank = new Bank();  
        SavingsAccount savingsAccount = new SavingsAccount(1000);  
        CurrentAccount currentAccount = new CurrentAccount(500);  
        bank.addAccount(savingsAccount);  
        bank.addAccount(currentAccount);  
        savingsAccount.deposit(200);  
        currentAccount.withdraw(100);  
        savingsAccount.calculateInterest();  
        currentAccount.calculateInterest();  
        bank.displayBalances();  
    }  
}
```

OUTPUT:

```
Balance: 1260.0  
Balance: 400.0
```