ASSIGNMENT:-5

QUESTION:-1

```
import java.util.Scanner;

abstract class Account
{
   protected String accountNumber;
   protected double accountBalance;

   Account(String accountNumber)
   {
        this.accountNumber=accountNumber;
             setAccountBalance(0);
   }

    public String getAccountNumber() {
        return accountNumber;
    }

    public double getAccountBalance() {
        return accountBalance;
    }

    public void setAccountNumber(String accountNumber) {
        this.accountNumber = accountNumber;
    }

    public void setAccountBalance(double accountBalance) {
        this.accountBalance = accountBalance;
    }


   public String toString()
   {
    return getAccountNumber()+"\t"+getAccountBalance();
   }

   abstract double computeInterest(int a);
}

class Checking extends Account
{
public double interest;
Checking(String accountNo){
super(accountNo);
}

public String toString()
{
    return getAccountNumber()+"\t"+getAccountBalance();
}

double computeInterest(int n){
                if((int)getAccountBalance()>700)
                {
                    interest=0.02*(getAccountBalance()-700)*n;
                }
            return interest;
    }
}

class Savings extends Account{

    private double rate;
    public double interest;
    Savings(String accountNo,double rate){
        super(accountNo);
```

```
        setRate(rate);
    }

    public double getRate() {
        return rate;
    }

    public void setRate(double rate) {
        if(rate<0)
            System.out.println("Negetive rate will not allowed");
        else
            this.rate = rate;
    }

    public String toString(){
        return getAccountNumber()+"\t"+getAccountBalance()+"\t"+getRate();
    }

    double computeInterest(int n) {
         interest=Math.pow((1+rate),n)*getAccountBalance()-getAccountBalance();
        return interest;
    }

}
public class ab24510_A5_1 {
    public static void main(String[] args) {
        Savings sav[]=new Savings[10];
        Checking ch[]=new Checking[10];
        Scanner sc=new Scanner(System.in);
        while(true){
        System.out.println("Select an Option\n1:To store 10 object of Savings class\n2:To
        store 10 object of Checking class\n3:To store 5 objects of Savings and 5 object of
        Checking class\n4:To exit");
        switch(sc.nextInt()){
            case 1:
                    for(int i=0;i<10;i++){
                    System.out.println("Enter account No. and initial balance");
                    sav[i]=new Savings(sc.next(),0.02);
                    sav[i].setAccountBalance(sc.nextDouble());
                    sav[i].interest=sav[i].computeInterest(3);
                    }
                    System.out.println("Account Number\tAccount Balance\tinterest
                    Rate\tinterest amount");
                    for(int i=0;i<10;i++){
                     System.out.println(sav[i].toString()+"\t"+(float)sav[i].interest);
                    }
                    break;
            case 2:
                    for(int i=0;i<10;i++){
                    System.out.println("Enter account No. and initial balance");
                    ch[i]=new Checking(sc.next());
                    ch[i].setAccountBalance(sc.nextDouble());
                    ch[i].interest=ch[i].computeInterest(3);
                    }
                    System.out.println("Account Number\tAccount Balance\tinterest amount");
                    for(int i=0;i<10;i++){
                     System.out.println(ch[i].toString()+"\t"+(float)ch[i].interest);
                    }
                    break;
            case 3:
                     int i=0;
                     for(i=0;i<5;i++){
                     System.out.println("Enter account No. and initial balance");
                     sav[i]=new Savings(sc.next(),0.02);
                     sav[i].setAccountBalance(sc.nextDouble());
                     sav[i].interest=sav[i].computeInterest(3);
                        }
                    for(int j=0;j<5;j++,i++){
                      System.out.println("Enter account No. and initial balance");
                       ch[j]=new Checking(sc.next());
                       ch[j].setAccountBalance(sc.nextDouble());
```

```
                        ch[j].interest=ch[j].computeInterest(3);
                        sav[i]=new Savings(ch[j].getAccountNumber(),0.02);
                        sav[i].setAccountBalance(ch[j].getAccountBalance());
                        sav[i].interest=ch[j].interest;
                         }
                    System.out.println("Account Number\tAccount Balance\tinterest
                    Rate\tinterest amount");
                    for(i=0;i<10;i++){
                     System.out.println(sav[i].toString()+"\t"+(float)sav[i].interest);
                    }
                    break;
            case 4:
                return;
            default:
                System.out.println("Invalid Option");
          }
        }
    }
}
```

QUESTION:-2

```
import java.io.FileInputStream;
import java.util.Random;
import java.util.Scanner;

class Player
{
    private String firstName;
    private String lastName;
    private int score;

    public String getFirstName()
    {
        return firstName;
    }

    public String getLastName()
    {
        return lastName;
    }

    public int getScore()
    {
        return score;
    }

    public void setFirstName(String firstName)
    {
        this.firstName = firstName;
    }

    public void setLastName(String lastName)
    {
        this.lastName = lastName;
    }

    public void setScore(int score)
    {
        this.score = score;
    }
}

abstract class Question
{
   private String question;
   private int points;
   private String answer;
   public int count;
    public String getAnswer() {
```

```
        return answer;
    }

    public void setAnswer(String answer) {
        this.answer = answer;
    }

   public String getQuestion()
   {
        return question;
   }
   public int getPoints()
   {
        return points;
   }
   public void setQuestion(String question)
   {
        this.question = question;
   }
   public void setPoints(int points)
   {
        this.points = points;
   }

   abstract public void read(FileInputStream fin, Player p);
}
class QuestionTF extends Question
{
    public void read(FileInputStream fin, Player p)
    {
        try
        {
            int x=0;
            char c='a';
            String temp="";
            while(true)
            {
               x=fin.read();
               c = (char)x;
               if(c=='\n')
                   break;
               temp = temp+""+c;
            }
            temp = temp.trim();
            setPoints(Integer.parseInt(temp));
            temp="";
            x=fin.read();
            c=(char)x;
            while(c!='\n'&&x>0)
            {
                temp=temp+""+c;
                x = fin.read();
                c=(char)x;
            }
            temp = temp.trim();
            setQuestion(temp);
            temp="";
            x=fin.read();
            c=(char)x;
            while(c!='\n'&&x>0)
            {
                temp=temp+""+c;
                x = fin.read();
                c=(char)x;
            }
            temp = temp.trim();
            setAnswer(temp);
        }
        catch (Exception e)
        {
            System.err.println(""+e);
```

```
        }
    }
    public void check()
    {

    }
}
class QuestionMC extends Question
{

    public void read(FileInputStream fin, Player p)
    {
        try
        {
            int x=0;
            char c='a';
            String temp="";
            while(true)
            {
               x=fin.read();
               c = (char)x;
               if(c=='\n')
                   break;
               temp = temp+""+c;
            }
            temp = temp.trim();
            setPoints(Integer.parseInt(temp));
            temp="";
            x=fin.read();
            c=(char)x;
            while(c!='\n')
            {
                temp=temp+""+c;
                x = fin.read();
                c=(char)x;
            }
            temp = temp.trim();
            setQuestion(temp);
            temp="";
            x=fin.read();
            c=(char)x;
            while(c!='\n'&&x>0)
            {
                temp=temp+""+c;
                x = fin.read();
                c=(char)x;
            }
            temp = temp.trim();
            String option="";
            int n = Integer.parseInt(temp);
            for(int i=0;i<n;i++)
            {
                temp="";
                x=fin.read();
                c=(char)x;
                while(c!='\n'&&x>0)
                {
                    temp=temp+""+c;
                    x = fin.read();
                    c=(char)x;
                }
                option =  option+"\n("+(char)(65+i) +") "+ temp;
            }
            //System.out.println(getQuestion()+""+option);
            setQuestion(getQuestion()+""+option+"\n");
            temp="";
            x=fin.read();
            c=(char)x;
            while(c!='\n'&&x>0)
            {
                temp=temp+""+c;
```

```
                x = fin.read();
                c=(char)x;
            }
            temp = temp.trim();
            setAnswer(temp);
        }
        catch (Exception e)
        {
            System.err.println(""+e);
        }
    }
    public void check()
    {

    }

}
class QuestionSA extends Question
{
    public void read(FileInputStream fin, Player p)
    {
        try
        {
            int x=0;
            char c='a';
            String temp="";
            while(true)
            {
               x=fin.read();
               c = (char)x;
               if(c=='\n')
                   break;
               temp = temp+""+c;
            }
            temp = temp.trim();
            setPoints(Integer.parseInt(temp));
            temp="";
            x=fin.read();
            c=(char)x;
            while(c!='\n')
            {
                temp=temp+""+c;
                x = fin.read();
                c=(char)x;
            }
            temp = temp.trim();
            setQuestion(temp);
            temp="";
            x=fin.read();
            c=(char)x;
            while(c!='\n'&&x>0)
            {
                temp=temp+""+c;
                x = fin.read();
                c=(char)x;
            }
            temp = temp.trim();
            setAnswer(temp);
        }
        catch (Exception e)
        {
            System.err.println(""+e);
        }
    }
    public void check()
    {

    }
}
class QuizBowl
{
```

```
   public static void main(String args[])
   {
       try
       {
           Player p = new Player();
           System.out.println("What is your first name?");
           Scanner scan= new Scanner(System.in);
           p.setFirstName(scan.next());
           System.out.println("What is your last name?");
           p.setLastName(scan.next());
           System.out.println("What files stores your Questions?");
           String file= scan.next();
           FileInputStream fin = new FileInputStream("C:\\Users\\HP-PC\\Desktop\\A5\\"+file);
           int x=0;
           String nQue="";
           try {
            while(true)
           {
               x=fin.read();
               if((char)x=='\n')
                   break;
               char c = (char)x;
//               System.out.print(c);
               nQue= nQue+""+c;
           }
           } catch (Exception e) {
               System.err.println(""+e);
           }
           nQue= nQue.trim();
           int n = Integer.parseInt(nQue);
           int queN=0;
           //System.out.println("HELLLEOO");
           while(true)
           {
               try
               {
                  System.out.println("How Many questions whould you like to (out of "+n+")");
                  queN=scan.nextInt();
                  if(queN>n)
                  {
                      System.out.println("Sorry, thats too many!!");
                      queN=0;
                  }
                  else if(queN<=0)
                  {
                      System.out.println("Sorry, not zero atleast or negative!!");
                      queN=0;
                  }
                  else
                  {
                      break;
                  }
                }
                catch (Exception e)
                {
                    System.out.println("Sorry, That is not valid");
                    break;
                }
           }
           Question que[] = new Question[Integer.parseInt(nQue)];
           int i=0;
           int k=1;
           while(x>=0)
           {
               x= fin.read();
               String qType = ""+(char)x+""+(char)fin.read()+""+(char)fin.read();
               if(qType.equals("MC "))
               {
                   que[i++]= new QuestionMC();
                   que[i-1].read(fin, p);
               }
```

```
                else if(qType.equals("TF "))
                {
                    que[i++]= new QuestionTF();
                    que[i-1].read(fin, p);
                }
                else if(qType.equals("SA "))
                {
                    que[i++]= new QuestionSA();
                    que[i-1].read(fin, p);
                }
                k++;
            }
            for(int j=0;j<queN;j++)
            {
                Random r=new Random();
                int randomCount = r.nextInt(n);
                if(que[randomCount].count==0)
                {
                 System.out.println("Point :- "+ que[randomCount].getPoints());
                 System.out.println("Question :- "+que[randomCount].getQuestion());
                 String reply=scan.next();
                 String temp = que[randomCount].getAnswer();
                 if(reply.equals(temp))
                  {
                      System.out.println("Correct ! You get "+que[randomCount].getPoints()+"
                      points");
                      p.setScore(que[randomCount].getPoints()+p.getScore());
                  }
                  else if(reply.toLowerCase().equals("skip"))
                  {
                      System.out.println("Okay! You Can Skip this question");
                      p.setScore(p.getScore()+0);
                  }
                  else
                  {
                      System.out.println("InCorrect ! You lost
                      "+que[randomCount].getPoints()+" points");
                      p.setScore(p.getScore()-que[randomCount].getPoints());
                  }
                 que[randomCount].count=1;
                }
                else
                {
                    j--;
                }
               }
            System.out.println(p.getFirstName()+" "+p.getLastName()+", your game is over!");
            System.out.println("Your final Score is "+p.getScore()+" points.");
            System.out.println("Better Luck next time!");
        }
         catch(Exception e)
         {

         }
    }
}
```