AM Computing Task 3 Specimen Marking Scheme

`` public class Member { protected int memberNo; protected String surname; protected String name; protected String phone; public Member(int memberNo, String surname, String name, String phone) { this.memberNo= memberNo; this.surname = surname; this.name = name; this.phone = phone; } public String toString() { return memberNo+" "+surname+" "+ name+" "+phone; } public void changePhone(String phone) { this.phone = phone; } public int getMemberNo() { return memberNo; } public String getSurname() { return surname; } public String getName() { return name; } public String getMobile() { return phone; } public void setName(String name) { this.name = name; } public void setSurName(String surname) { this.surname = surname; } public void setMobile (String phone) { this.phone = phone; } } ```	Award 3 marks for the correct use of getters and setters Award 2 marks for the declaration and correct use of constructors (default and/or parameterised) Award 1 mark for the correct use of toString() method (for member, JuniorMember and SeniorMember classes)	6
`` import java.util.Scanner; public class Junior_Member extends Member { private String type; public Junior_Member() { super(0,null,null,null); type = null; ```	Award 1 mark for the use of extends to inherit the member class Award 2 marks for the use of super to invoke the member class constructor	20

```
    }

    public Junior_Member(int memberNo, String name, String
surname, String phone)
    {
        super(memberNo,surname,name,phone);
        this.type = "Junior";
    }

     public void input(Junior_Member[]JuniorArray)
     {
        Scanner sc = new Scanner(System.in);
        for (int k=0; k<JuniorArray.length; k++)
        {
        System.out.println("Member id ");
        int id = sc.nextInt();
        System.out.println("Member name ");
        String name = sc.next();
        System.out.println("Member Surname ");
        String surname = sc.next();
        System.out.println(" Phone number ");
        String phone = sc.next();

        JuniorArray[k] = new
Junior_Member(id,name,surname,phone);
    }
    }

      public void
displayJuniorMembers(Junior_Member[]JuniorArray )
    {
        for (int k=0; k< JuniorArray.length; k++)
        {
            System.out.println(JuniorArray[k]);
        }
    }

    public int searchMember(Junior_Member[]JuniorArray,int
memNo)
     {
         int posi = -1;
         for (int k=0; k< JuniorArray.length; k++)
        {
            if (JuniorArray[k].getMemberNo() == memNo)
                   posi = k;
        }
        return posi;
    }

    public void editJunior(Junior_Member[]JuniorArray, int
posi)
    {
        Scanner sc = new Scanner(System.in);
        int tempMemNo       =
JuniorArray[posi].getMemberNo();
        String tempName     = JuniorArray[posi].getName();
        String tempSurname  =
JuniorArray[posi].getSurname();
        String tempMobile   =
JuniorArray[posi].getMobile();

        System.out.println("Member details to edit
"+JuniorArray[posi]);
        int choice2;
        do {
              System.out.println("[1]..Edit Name  ");
              System.out.println("[2]..Edit SurName ");
              System.out.println("[3]..Edit Mobile No ");
              System.out.println("[4]..Return to Main Menu
");
              choice2 = sc.nextInt();
              switch (choice2) {
                  case 1 : System.out.println("New Name
");
                           tempName = sc.next();
```

Award 2 marks for the use of polymorphism
Award 1 mark for the use of encapsulation
Award 2 marks for the declaration and correct use of constructors (default and/or parameterised)
Award 1 mark for the use of constructor overloading
Award 1 mark for the use of toString() method overriding
Award 2 marks for the correct use of array of objects
Award 1 mark for the correct use of syntax to input a senior and a junior member
Award 1 mark for the correct use of syntax to display member details
Award 2 marks for the correct use of syntax to search for a particular member
Award 4 marks for the functionality to edit a member by name, surname and mobile number

```
                            break;

                    case 2 : System.out.println("New SurName
");
                             tempSurname = sc.next();
                             break;

                    case 3 : System.out.println("New Mobile
No. ");
                             tempMobile = sc.next();
                             break;

                    case 4 : JuniorArray[posi] =
                                 new Junior_Member(tempMemNo
,tempName ,tempSurname,tempMobile);
                              break;
                  }// choice2

                } while (choice2 != 4);

            }

    public String toString()
    {
        return super.toString()+"  "+type ;
    }

}


import java.util.Scanner;
public class Senior_Member extends Member
{
    private String type;

    public Senior_Member()
    {
        super(0,null,null,null);
        type = null;
    }

    public Senior_Member(int memberNo, String name, String
surname, String phone)
    {
        super(memberNo,surname,name,phone);
        this.type = "Senior";
    }

    public void input(Senior_Member[]SeniorArray )
    {
        Scanner sc = new Scanner(System.in);
        for (int k=0; k< SeniorArray.length; k++)
        {
        System.out.println("Member id ");
        int id = sc.nextInt();
        System.out.println("Member name ");
        String name = sc.next();
        System.out.println("Member Surname ");
        String surname = sc.next();
        System.out.println(" Phone number ");
        String phone = sc.next();
        SeniorArray[k] = new
Senior_Member(id,name,surname,phone);
       }

    }

    public void
displaySeniorMembers(Senior_Member[]SeniorArray )
    {
        for (int k=0; k< SeniorArray.length; k++)
        {
            System.out.println(SeniorArray[k]);
        }
```

```
    }

     public int
searchMember(Senior_Member[]SeniorArray,int memNo)
     {
         int posi = -1;
         for (int k=0; k< SeniorArray.length; k++)
        {
            if (SeniorArray[k].getMemberNo() == memNo)
                   posi = k;

        }

        return posi;
    }

    public void editSenior(Senior_Member[]SeniorArray, int
posi)
    {
        Scanner sc = new Scanner(System.in);
        int tempMemNo       =
SeniorArray[posi].getMemberNo();
        String tempName     = SeniorArray[posi].getName();
        String tempSurname  =
SeniorArray[posi].getSurname();
        String tempMobile   =
SeniorArray[posi].getMobile();

        System.out.println("Member details to edit
"+SeniorArray[posi]);
        int choice2;
        do {
              System.out.println("[1]..Edit Name  ");
              System.out.println("[2]..Edit SurName ");
              System.out.println("[3]..Edit Mobile No ");
              System.out.println("[4]..Return to Main Menu
");
              choice2 = sc.nextInt();
              switch (choice2) {
                  case 1 : System.out.println("New Name
");
                           tempName = sc.next();

                           break;

                  case 2 : System.out.println("New SurName
");
                           tempSurname = sc.next();
                           break;

                  case 3 : System.out.println("New Mobile
No. ");
                           tempMobile = sc.next();
                           break;

                  case 4 : SeniorArray[posi] =
                               new Senior_Member(tempMemNo
,tempName ,tempSurname,tempMobile);
                            break;
                }// choice2

               } while (choice2 != 4);
    }

    public String toString()
    {
        return super.toString()+"  "+type ;
    }
}
```

`import java.util.Scanner;` `public class Use_Members` `{` `  public static void main (String args[])`	Award 2 marks for the creation of array of Objects for the Senior and Junior members	7

```
    {
        Scanner sc = new Scanner(System.in);
        Senior_Member[] s = new Senior_Member[2];
        Junior_Member[] j = new Junior_Member[2];


        Senior_Member sm = new Senior_Member();
        Junior_Member jm = new Junior_Member();
        int choice, choice2;
        String name, surname, mobile;

        do {
            System.out.println("[1]..Enter Senior Member
");
            System.out.println("[2]..Enter Junior Member
");
            System.out.println("[3]..Display Senior
Members ");
            System.out.println("[4]..Display Junior
Members ");
            System.out.println("[5]..Edit Senior Member
");
            System.out.println("[6]..Edit Junior Member
");
            System.out.println("[7]..Exit ");
            System.out.println("Make your choice ");
            choice = sc.nextInt();
            int memNo= -999, posi;

            switch (choice){
                case 1: sm.input(s);
                        break;
                case 2: jm.input(j);
                        break;
                case 3:
                        sm.displaySeniorMembers(s);
                        break;
                case 4:
                        jm.displayJuniorMembers(j);
                        break;
                case 5:
                // reduction of redundant code by taking
into consideration
               // whether it is a senior or junior member.
                case 6: System.out.println("Enter Member
number ");
                        memNo = sc.nextInt();

                         if(choice == 5)
                            {
                             posi =
sm.searchMember(s,memNo);
                              sm.editSenior(s,  posi);

                            }
                         else
                            {
                                posi =
jm.searchMember(j,memNo);
                                jm.editJunior(j,posi);
                            }
                    }  // choice

                }while (choice != 7);
            }
        }
```

Award
Award 1 mark for the creation of Objects for the Senior and Junior members
Award 2 marks for the correct use of loop construct for the Main Menu
Award 1 mark for the correct use of if or switch statements within the Main Menu
Award 1 mark for calling of methods with or without arguments

	Award 3 marks for correct use of syntax Award 1 mark for the correct and meaningful use of naming standards for classes, methods and variables	7

	Award 1 mark for the declaration and correct use of variables Award 2 marks for the correct use of access modifiers (public, private and protected)	
	Total	40