INTERNATIONAL SECONDARY EXAMINATION
NOVEMBER 2023

COMPUTER SCIENCE: PAPER I

MARKING GUIDELINES

Time: 3 hours 150 marks

These marking guidelines are prepared for use by examiners and sub-examiners, all of whom are required to attend a standardisation meeting to ensure that the guidelines are consistently interpreted and applied in the marking of candidates' scripts.

The IEB will not enter into any discussions or correspondence about any marking guidelines. It is acknowledged that there may be different views about some matters of emphasis or detail in the guidelines. It is also recognised that, without the benefit of attendance at a standardisation meeting, there may be different interpretations of the application of the marking guidelines.

SECTION A SQL

QUESTION 1

Question 1.1 (4)

Access/MySQL/JavaDB:
```
SELECT *
FROM tblDonors
WHERE Individual = true
ORDER BY DonorName
```

Question 1.2 (6)

Access/MySQL:
```
SELECT CentreName, City, ROUND (NumChildren/NumStaff,0) AS Ratio
FROM tblCentres
WHERE ROUND(NumChildren/NumStaff,0) > 30
```

JavaDB:
```
SELECT CentreName, City, FLOOR (NumChildren/cast(NumStaff AS REAL)
+ 0.5) AS Ratio
FROM tblCentres
WHERE FLOOR(NumChildren/cast(NumStaff AS REAL) + 0.5)> 30
```

Question 1.3 (4)

Access/MySQL/JavaDB:
```
SELECT MIN(Amount), MAX(Amount)
FROM tblDonations
WHERE YEAR(DonationDate) = 2021
```

OR Access:
```
SELECT MIN(Amount), MAX(Amount)
FROM tblDonations
WHERE DonationDate BETWEEN #2021/01/01# AND #2021/12/31#
```

OR Access
```
SELECT MIN(Amount), MAX(Amount)
FROM tblDonations
WHERE DonationDate> #2021/01/01# AND DonationDate < #2021/12/31#
```

OR MySQL:
```
SELECT MIN(Amount), MAX(Amount)
FROM tblDonations
WHERE DonationDate BETWEEN '2021/01/01' AND '2021/12/31'
```


OR

```
SELECT MIN(Amount), MAX(Amount)
FROM tblDonations
WHERE DonationDate> ‘2021/01/01’ AND DonationDate < ‘2021/12/31’
```

OR JavaDB:

```
SELECT MIN(Amount), MAX(Amount)
FROM tblDonations
WHERE DonationDate> ‘2021-01-01’ AND DonationDate < ‘2021-12-31’
```

OR JavaDB

```
SELECT min(amount), max(amount)
FROM tblDonations
WHERE DonationDate BETWEEN '2021-01-01' AND '2021-12-31';
```

Question 1.4 (6)

```
SELECT DonorName, Amount, CertificateIssued
FROM tblDonors, tblDonations
WHERE tblDonations.DonorID = tblDonors.DonorID AND
(CertificateIssued = FALSE OR Amount > 90000)
```

OR

```
SELECT DonorName, Amount, CertificateIssued
FROM tblDonors INNER JOIN tblDonations
ON tblDonations.DonorID = tblDonors.DonorID
WHERE (CertificateIssued = FALSE OR Amount > 90000)
```

Question 1.5 (7)

```
SELECT DISTINCT City
FROM tblCentres
WHERE City NOT IN
(SELECT City FROM tblCentres, tblDonations WHERE
tblCentres.CentreID = tblDonations.CentreID AND DonorID = 19)
```

Note: looking for NULL values as a result of a LEFT JOIN will not give the correct answer.

Java DB

```
SELECT DISTINCT CITY
FROM tblCentres
WHERE CITY NOT IN (SELECT City FROM tblCentres
LEFT JOIN tblDonations ON
tblCentres.CentreID=tblDonations.CenterID
WHERE DONORID=19)
```

Question 1.6 (8)

Access:

```
SELECT CentreName, DonorName, COUNT(*)
FROM tblCentres, tblDonors, tblDonations
WHERE tblCentres.CentreID = tblDonations.CentreID AND
tblDonors.DonorID = tblDonations.DonorID
GROUP BY CentreName, DonorName
HAVING Count(*) > 2
```

OR

```
SELECT CentreName, DonorName, COUNT(*)
FROM tblCentres INNER JOIN (tblDonors INNER JOIN tblDonations
ON tblDonors.DonorID = tblDonations.DonorID) ON
tblCentres.CentreID = tblDonations.CentreID
GROUP BY CentreName, DonorName
HAVING Count(*) > 2
```

Question 1.7 (9)

Access:

```
UPDATE tblDonors SET Contact = LEFT (Contact,LEN(Contact) -
8)&"yahoo.com"
WHERE Contact LIKE "*yaho.com"
```

MySQL:

```
UPDATE tblDonors SET Contact = CONCAT (LEFT
(Contact,LENGTH(Contact) -8),'yahoo.com')
WHERE Contact LIKE '%yaho.com'
```

JavaDB:

```
UPDATE tblDonors SET Contact = SUBSTR (Contact,1 ,LENGTH(Contact) -
8) || 'yahoo.com'
WHERE Contact LIKE '%yaho.com'
```

Question 1.8 (6)

Access:

```
INSERT INTO tblCentres (CentreName, City, NumStaff, NumChildren)
SELECT "Adult " & CentreName, City, NumStaff, 0
FROM tblCentres
WHERE City = "Windhoek"
```

MySQL:

```
INSERT INTO tblCentres (CentreName, City, NumStaff, NumChildren)
SELECT CONCAT('Adult ', CentreName), City, NumStaff, 0
FROM tblCentres
WHERE City = 'Windhoek'
```


JavaDB:

```
INSERT INTO tblCentres (CentreName, City, NumStaff, NumChildren)
SELECT 'Adult ' || CentreName, City, NumStaff, 0
FROM tblCentres
WHERE City = 'Windhoek'
```


SECTION B

JAVA SOLUTION

QUESTION 2

```
//Question 2
//Question 2.1 - 3
//class header
//all fields private
//all correctly typed with correct names
public class Member {

    private String memberName;
    private LocalDate dob;
    private int foodPref;
    private String medAidName;

    //Question 2.2 - 2
    //constant declared with final / constant and named and typed
    correctly
    //assigned correct value
    public static final int FOOD_STANDARD = 0;
    public static final int FOOD_HALAAL = 1;
    public static final int FOOD_KOSHER = 2;
    public static final int FOOD_VEGETARIAN = 3;

    //Question 2.3 - 4
    //correct header
    //correct parameter names and types
    //fields set to parameters, deduct 1 per mistake, max of
    2 deductions

public Member(String inMemberName, LocalDate inDob, int inFoodPref,
    String inMedAidName) {
        memberName = inMemberName;
        dob = inDob;
        foodPref = inFoodPref;
        medAidName = inMedAidName;
    }

    //Question 2.4 - 2
    //correct header
    //correct return statement
    public String getMemberName() {
        return memberName;
    }
```

```
//Question 2.5 - 5
//correct header
//switch OR if else structure (see alternate solution) on foodPref variable
//use of constants instead of numbers
//correct return statements
//return "other" as default or final else

public String getFoodPref() {
    switch (foodPref) {
        case FOOD_STANDARD:
            return "standard";
        case FOOD_HALAAL:
            return "halaal";
        case FOOD_KOSHER:
            return "kosher";
        case FOOD_VEGETARIAN:
            return "vegetarian";
        default:
            return "other";
    }
}

ALTERNATE SOLUTION:
public String getFoodPref() {
    if (foodPref == FOOD_STANDARD) {
        return "standard";
    } else if (foodPref == FOOD_HALAAL) {
        return "halaal";
    } else if (foodPref == FOOD_KOSHER) {
        return "kosher";
    } else if (foodPref == FOOD_VEGETARIAN) {
        return "vegetarian";
    } else {
        return "other";
    }
}
```



```
//Question 2.6 - 4
    //correct header
    //comparison between LocalDate and now()
    //correct age returned, taking into account whether birthday has passed or not
    //return age
    public int getAge() {
        Period age = Period.between(dob, LocalDate.now());
        return age.getYears();
    }
    ALTERNATE SOLUTION:
    public int getAge() {
        int yearNow = LocalDate.now().getYear();
        int yearDob = dob.getYear();
        int age = yearNow - yearDob;
        if (LocalDate.now().getMonthValue()< dob.getMonthValue()) {
            age--;
        } else if (LocalDate.now().getMonth() == dob.getMonth() &&
    LocalDate.now().getDayOfMonth()< dob.getDayOfMonth()) {
            age--;
        }
        return age;
    }

    //Question 2.7 - 4
    //correct header
    //contains all fields
    //use of getAge() and getFoodPref()
    //return formatted string
    public String toString() {
        return memberName + "\t" + getAge() + " years old\tFood: "
    + getFoodPref() + "\tMedical Aid: " + medAidName;
    }

}
```

QUESTION 3

```
//Question 3
//Question 3.1 - 4
//class header
//class inherits from Unsponsored class
//all fields private
//all fields correctly typed with correct names
public class SponsoredMember extends Unsponsored {

    private String sponsorName;
    private int amount;

    //Question 3.2 - 5
    //constructor named correctly
    //parameters correct
    //super method called
    //inMemberName, inDob, inFoodPref, inMedAidName sent as
    parameters to super method
    //parameters assigned


    public SponsoredMember(String inMemberName, LocalDate inDob,
    int inFoodPref, String inMedAidName, String inSponsorName, int
    inAmount) {

        super(inMemberName, inDob, inFoodPref, inMedAidName);

        sponsorName = inSponsorName;

        amount = inAmount;

    }
    //Question 3.3 - 2
    //correct headers for accessors
    //correct returns for accessors
    public boolean getSponsorName() {
        return amountsponsorName;
    }

    public int getAmount() {

        return amount;

    }
```

```
//Question 3.4 - 3
    //correct header
    //call to super method
    //return correctly formatted string


    public String toString() {

       return super.toString() + "\tSponsor Name: " + sponsorName +
    "\tAmount: " + amount;

   }}
```


QUESTION 4

```
//Question 4
//Question 4.1 - 1
//class created correctly
public class MemberManager {

    //Question 4.2 - 4
    //both properties private
    //Member array declared with correct name
    //array size set to 100
    //size property created

    private Member[] mArr = new Member[100];
    private int size = 0;
     ;
    //Question 4.3 - 13
    //constructor header correct
    //try catch implemented correctly to handle exception
    //open file for reading
    //loop through all lines
    //read next line from file
    //split line into required parts
    //parse date into LocalDate
    //if line contains additional data for SponsoredMember
    //extract additional data for SponsoredMember
    //create SponsoredMember object when necessary
    //create Member object when necessary
    //add created object to array
    //increment size

     public MemberManager() {
            try {
                Scanner scFile = new Scanner(new
    File("members.txt"));
                while (scFile.hasNextLine()) {
                    Scanner scLine = new
    Scanner(scFile.nextLine()).useDelimiter("#");
                    String memberName = scLine.next();
                    LocalDate dob = LocalDate.parse(scLine.next);
                    int foodPref = scLine.nextInt();
                    String medAidName = scLine.next();
                    if (scLine.hasNext()) {
                        String sponsorName = scLine.next();
                        int amount = scLine.nextInt();
                        mArr[size] = new
    SponsoredMember(memberName, dob, foodPref, medAidName,
    sponsorName, amount);
                    } else {
                        mArr[size] = new Member(memberName, dob,
    foodPref, medAidName);
                    }
                    size++;
```

```
                }
            } catch (FileNotFoundException ex) {
                System.out.println("Error, file not found.");
            }
        }

    //Question 4.4 - 5
    //method header correct
    //string initialised
    //loop through Member array
    //append to string
    //return string
    public String toString() {
            String ret = "";
            for (int i = 0; i < size; i++) {
                ret += mArr[i] + "\n";
            }
            return ret;
    }
    //Question 4.5 - 5
    //method header correct
    //loop through Member array
    //if inside loop to check if current object name equals
    parameter name
    //return age using getAge() method
    //return -1 outside loop

    public int ageOfMember(String inName) {
        for (int i = 0; i < size; i++) {
            if (mArr[i].getName().equals(inName)) {
                return mArr[i].getAge();
            }
        }
        return -1;
    }
```


QUESTION 6.1 & 6.2

```
//Question 6.1 - 2
//create centre amount as a global class constant
//assign 40 to the centre amount constant
//could be any variable name, must be capitalised
private final int CENTRE_AMOUNT = 40;


//Question 6.2 - 8
//create 3 variables to hold amount totals
//loop through array
//if inside loop to check whether current object is a
SponsoredMember child
//cast current object as SponsoredMember
//nested if to check whether current object is fully sponsored
or needs additional money from centre (may hardcode 40 or use
CENTRE_AMOUNT
//increment entirelySponsored / sponsoredAndCentre in correct
positions
//else on outer if statement to catch entirelyCentre members
//return correctly formatted statement of totals


public String totalMemberTypes() {
   int entirelyCentre = 0;
   int entirelySponsored = 0;
   int sponsoredAndCentre = 0;
   for (int i = 0; i < size; i++) {
       if (mArr[i] instanceof SponsoredMember) {
           SponsoredMember temp = (SponsoredMember) mArr[i];
           if (temp.getAmount() >= CENTRE_AMOUNT) {
               entirelySponsored++;
           } else {
               sponsoredAndCentre++;
           }
       } else {
           entirelyCentre++;
       }
   }
   return "Entirely Centre: " + entirelyCentre + "\nEntirely
Sponsored: " + entirelySponsored + "\nSponsored and Centre: " +
sponsoredAndCentre + "\n";
                                                              }
```

QUESTION 7.1

```
//Question 7.1 - 16
//correct header with parameter
//centreSum initialised to 0 outside of file loop
//loop through entire file using Scanner
//extract sponsorName from file inside loop
//sponsoredSum initialised to 0 inside of file loop
//for loop to go through entire array
//if inside for loop to check if current object is a
SponsoredMember
//cast current object to SponsoredMember
//nested if to check if current object's sponsor equals the
current sponsorName from the text file
//nested if to see if current object's amount is >=
CENTRE_AMOUNT (may not hardcode 40).
//add CENTRE+AMOUNT to current sponsorName's total (do not
penalise again for use of 40).
//add sponsored amount using getAmount() in else portion of
nested if.
//add any shortfall to centre total (do not penalise again for
use of 40).
//for loop to run through whole array after file loop has
terminated
//add CENTRE_AMOUNT (do not penalise again for use of 40) to
//total for centre for all unsponsored members
//return the correct string


public String spendingMoneyReport(String filename) {
        String ret = "Spending Money Report\n";
        int centreSum = 0;
        try {
            Scanner scFile = new Scanner(new File(filename));
            while (scFile.hasNextLine()) {
                String sponsorName = scFile.nextLine();
                int sponsorSum = 0;
                for (int i = 0; i < size; i++) {
                    if (mArr[i] instanceof SponsoredMember) {
                        SponsoredMember temp =
(SponsoredMember) mArr[i];
                        if
(sponsorName.equals(temp.getSponsorName())) {
                            if (temp.getAmount() >=
CENTRE_AMOUNT) {
                                sponsorSum = sponsorSum +
CENTRE_AMOUNT;
                            } else {
                                sponsorSum = sponsorSum +
temp.getAmount();
                                centreSum = centreSum +
(CENTRE_AMOUNT - temp.getAmount());
                            }
```

```
                        }
                    }
                }
                ret = ret + sponsorName + " total = $" +
 sponsorSum + "\n";
            }
            for (int i = 0; i < size; i++) {
                if (!(mArr[i] instanceof SponsoredMember)) {
                    centreSum = centreSum + CENTRE_AMOUNT;
                }
            }
            ret = ret + "Centre Total = $" + centreSum + "\n";
        } catch (FileNotFoundException ex) {
            System.out.println("Error. File not found.");
        }
        return ret;
}
```

QUESTION 5 & 6.3 & 7.2

```
//Question 5
//Question 5.1 - 1
//class created with main method
public class MemberUI {

    public static void main(String[] args) {
        //Question 5.2 - 1
        //MemberManager created in correct position
        MemberManager mm = new MemberManager();
        //Question 5.3 - 1
        //print MemberManager object
        System.out.println(mm);
        //Question 5.4 - 3
        //pass parameter "Stuart Redford"
        //call method ageOfMember correctly
        //method call inside println
        System.out.println(mm.ageOfMember("Stuart Redford"));""

        //Question 6.3 - 1
        //call totalMemberTypes correctly, do not penalise for lack of
        / incorrect println
        System.out.println(mm.totalMemberTypes());

        //Question 7.2 - 1
        //call spendingMoneyReport correctly, do not penalise for lack of / incorrect
        println
        //correct parameters sent to spendingMoneyReport method

        System.out.println(mm.spendingMoneyReport("Sponsors.txt"))
        ;    }
}
```

Total: 150 marks

DELPHI SOLUTION

QUESTION 2

```
//Question 2
//Question 2.1 - 3
//class header
//all fields private
//all correctly typed with correct names
unit Member;

interface
uses SysUtils, DateUtils;

type TMember = class
  private
    memberName : string;
    dob : TDateTime;
    foodPref : integer;
    medAidName : string;
    //Question 2.2 - 2
    //constant declared with final / constant and named and typed
    correctly
    //assigned correct value
    public
    const
      FOOD_STANDARD = 0;
      FOOD_HALAAL = 1;
      FOOD_KOSHER = 2;
      FOOD_VEGETARIAN  = 3;

      constructor Create(inMemberName : string; inDob : TDateTime;
                inFoodPref : integer; inMedAidName : string);
      function getMemberName() : string;
      function getFoodPref() : string;
      function getAge() : integer;
      function toString() : string;

end;

implementation

{ TMember }
```

```
//Question 2.3 - 4
//correct header
//correct parameter names and types
//fields set to parameters, deduct 1 per mistake, max of
    2 deductions

constructor TMember.Create(inMemberName: string; inDob: TDateTime;
    inFoodPref: integer; inMedAidName: string);
begin
    memberName := inMemberName;
    dob := inDob;
    foodPref := inFoodPref;
    medAidName := inMedAidName;
end;

    //Question 2.4 - 2
    //correct header
    //correct return statement
    function TMember.getMemberName: string;
    begin
      Result := memberName;
    end;
    //Question 2.5 - 5
    //correct header
    //switch OR if else structure (see alternate solution) on foodPref variable
    //use of constants instead of numbers
    //correct return statements
    //return "other" as default or final else

    function TMember.getFoodPref: string;
    begin
      case foodPref of
        FOOD_STANDARD : Result := 'standard';
        FOOD_HALAAL : Result := 'halaal';
        FOOD_KOSHER : Result := 'kosher';
        FOOD_VEGETARIAN : Result := 'vegetarian";'
        else Result := 'other';
      end;
    end;

    ALTERNATE SOLUTION:
    Accept nested if instead of case statement.
```



```
//Question 2.6 - 4
    //correct header
    //comparison between date and Now
    //correct age returned, taking into account whether birthday has passed or not
    //return age
    function TMember.getAge: integer;
    begin
      Result :=  1 + DateUtils.YearsBetween(Now, dob);
      if DateUtils.IncYear(dob, Result) > Now then
        dec (Result);
    end;

    ALTERNATE SOLUTION:
    Function TMember.getAge: integer;
    var
       yearNow, yearDob, monthNow, monthDob, dayNow, dayDob, age :
    integer;
    begin
       yearNow := Now.getYear();
       monthNow := Now.getMonth();
       dayNow := Now.getDay();
       yearDob := dob.getYear();
       monthDob := dob.getMonth();
       dayDob := dob.getDay();
       age := yearNow - yearDob;
       if monthNow < monthDob then
       begin
          age := age - 1;
       end
       else
       if (monthNow = dobMonth) AND (dayNow < dayMonth) then
       begin
          age := age - 1;
       end
       Result := age;
    end;

    //Question 2.7 - 4
    //correct header
    //contains all fields
    //use of getAge() and getFoodPref()
    //return formatted string
    function TMember.toString: string;
    begin
     Result := memberName + #9 + IntToStr(getAge()) + ' years old'
     + #9 + 'Food: ' + getFoodPref() + #9 + 'Medical Aid: ' +
     medAidName;
    end;
end.
```


QUESTION 3

```
//Question 3
//Question 3.1 - 4
//class header
//class inherits from Unsponsored class
//all fields private
//all fields correctly typed with correct names
unit SponsoredMember;

interface
uses SysUtils, DateUtils, Member;

type TSponsoredMember = class(TMember)
  private
    sponsorName : string;
    amount : integer;

  public
    constructor Create(inMemberName : string; inDob : TDateTime;
     inFoodPref : integer; inMedAidName : string; inSponsorName :
     string; inAmount : integer);
    function getSponsorName() : string;
    function getAmount() : integer;
    function toString() : string;
end;

     //Question 3.2 - 5
     //constructor named correctly
     //parameters correct
     //super method called
     //inMemberName, inDob, inFoodPref, inMedAidName sent as
     parameters to super method
     //parameters assigned
     implementation

{ TSponsoredMember }

constructor TSponsoredMember.Create(inMemberName : string; inDob :
    TDateTime; inFoodPref : integer; inMedAidName : string;
    inSponsorName : string; inAmount : integer);
begin
  Inherited Create (inMemberName, inDob, inFoodPref, inMedAidName);
  sponsorName := inSponsorName;
  amount := inAmount;
end;
```



```
     //Question 3.3 - 2
    //correct headers for accessors
    //correct returns for accessors
    function TSponsoredMember.getSponsorName: string;
    begin
      Result := sponsorName;
    end;

    function TSponsoredMember.getAmount: integer;
    begin
      Result := amount;
    end;

//Question 3.4 - 3
    //correct header
    //call to super method
    //return correctly formatted string
    function TSponsoredMember.toString: string;
    begin
     Result := Inherited toString + #9 + 'Sponsor Name: ' +
     sponsorName + #9 + 'Amount: ' + IntToStr(amount);
    end;

end.
```


QUESTION 4

```
//Question 4
//Question 4.1 - 1
//class created correctly
unit MemberManager;

interface
uses SysUtils, DateUtils, Member, SponsoredMember;

//Question 4.2 - 4
//both properties private
//Member array declared with correct name
//array size set to 100
//size property created

type TMembermanager = class
  private
    mArr : array[1..100] of TMember;
    size : integer;
  public
    const CENTRE_AMOUNT = 40;

    Constructor Create();
    function toString() : string;
    function ageOfMember(inName : string) : integer;
    function totalMemberTypes() : string;
    function spendingMoneyReport() : string;

end;

implementation
  //Question 4.3 - 13
  //constructor header correct
  constructor TMemberManager.Create;
  var
    inFile : textfile;
    line, memberName, medAidName, sponsorName, date, d, m, y :
     string;
    dob : TDateTime;
    foodPref, amount : integer;
 //check if file exists
 begin
    if FileExists('Members.txt') <> true then
      begin
        WriteLn('File not found.');
      end
    else
      begin
       //open file for reading
        AssignFile(inFile, 'Members.txt');
        Reset(inFile);
        size := 0;
      //loop through all lines
```

```
        while NOT EOF(inFile) do
          begin
           //read next line from file
            ReadLN(inFile, line);
           //increment size
            Inc(size);
           //split line into required parts
            memberName := Copy(line, 1, Pos('#', line) -1);
            Delete(line, 1, Pos('#', line));
            date := Copy(line, 1, Pos('#', line) -1);
            Delete(line, 1, Pos('#', line));
            y := Copy(date,1 , Pos('-', date) - 1);
            Delete(date,1 , Pos('-', date));
            m := Copy(date,1 , Pos('-', date) - 1);
            Delete(date,1 , Pos('-', date));
            d := date;
            WriteLn(y + ' ' + m + ' ' + d);
           //encode date correctly
            dob := EncodeDate(StrToInt(y), StrToInt(m),
    StrToInt(d));
           foodPref := StrToInt(Copy(line, 1, Pos('#', line) -1));
           Delete(line, 1, Pos('#', line));
          //if line contains additional data for SponsoredMember
           if Pos('#', line) > 0 then
             begin
               //extract additional data for SponsoredMember
               medAidName := Copy(line, 1, Pos('#', line) -1);
               Delete(line, 1, Pos('#', line));
               sponsorName := Copy(line, 1, Pos('#', line) -1);
               Delete(line, 1, Pos('#', line));
               amount := StrToInt(line);
               //create SponsoredMember object
               mArr[size] := TSponsoredMember.Create(memberName,
    dob, foodpref, medAidName, sponsorName, amount);
             end
           else
             begin
               medAidName := line;
               //create Member object when necessary
               //add created object to array
               mArr[size] := TMember.Create(memberName, dob,
    foodPref, medAidName);
             end;
         end;
      end;
  end;

    //Question 4.4 - 5
    //method header correct
    //string initialised
    //loop through Member array
    //append to string
    //return string
    function TMemberManager.toString: string;
```

```
  var
    i : integer;
    output : string;
  begin
    output := '';
    for i := 1 to size do
      begin
        output := output + mArr[i].toString() + #13#10;
      end;
    Result := output;
  end;

//Question 4.5 - 5
//method header correct
//loop through Member array
//if inside loop to check if current object name equals
parameter name
//return age using getAge() method
//return -1 outside loop

function TMemberManager.ageOfMember(inName: string): integer;
  var
    found : boolean;
    age : integer;
  begin
    found := false;
    age := -1;
    for i := 1 to size do
    begin
   if mArr[i].getMemberName() = inName then
      begin
        age := mArr[i].getAge();
      end;
    end;
    Result := age;
  end;
```

QUESTION 6.1 & 6.2

```
//Question 6.1 - 2
//create centre amount as a global class constant
//assign 40 to the centre amount constant
// could be any variable name, must be capitalised]
// NB: Should be in interface of TMemberManager
const CENTRE_AMOUNT = 40;

//Question 6.2 - 8
//create 3 variables to hold amount totals
//loop through array
//if inside loop to check whether current object is a
SponsoredMember child
//use AS TSponsoredMember at least once
//nested if to check whether current object is fully sponsored
or needs additional money from centre (may hardcode 40 or use
CENTRE_AMOUNT
//increment entirelySponsored / sponsoredAndCentre in correct
positions
//else on outer if statement to catch entirelyCentre members
//return correctly formatted statement of totals

function TMemberManager.totalMemberTypes: string;
  var
     i, entirelyCentre, entirelySponsored, sponsoredAndCentre :
 integer;
    output : string;
  begin
    output := '';
    entirelyCentre := 0;
    entirelySponsored := 0;
    sponsoredAndCentre := 0;
    for i := 1 to size do
    begin
      if mArr[i] IS TSponsoredMember then
      begin
         if (mArr[i] AS TSponsoredMember).getAmount() >=
 CENTRE_AMOUNT then
          begin
            entirelySponsored := entirelySponsored + 1;
          end
        else
          begin
            sponsoredAndCentre := sponsoredAndCentre + 1;
          end;
      end
      else
      begin
        entirelyCentre := entirelyCentre + 1;
      end;
  end;
```

```
    Result := 'Entirely Centre: ' + IntToStr(entirelyCentre) +
    #13#10 + 'Entirely Sponsored: ' + IntToStr(entirelySponsored) +
    #13#10 + 'Sponsored and Centre: ' +
    IntToStr(sponsoredAndCentre) + #13#10;
end;
```


QUESTION 7.1

```
    //Question 7.1 - 16
    //correct header with parameter
    //centreSum initialised to 0 outside of file loop
    //loop through entire file using Scanner
    //extract sponsorName from file inside loop
    //sponsoredSum initialised to 0 inside of file loop
    //for loop to go through entire array
    //if inside for loop to check if current object is a
    SponsoredMember
    //use AS TSponsoredMember at least once
    //nested if to check if current object's sponsor equals the
    current sponsorName from the text file
    //nested if to see if current object's amount is >=
    CENTRE_AMOUNT (may not hardcode 40).
    //add CENTRE+AMOUNT to current sponsorName's total (do not
    penalise again for use of 40).
    //add sponsored amount using getAmount() in else portion of
    nested if.
    //add any shortfall to centre total (do not penalise again for
    use of 40).
    //for loop to run through whole array after file loop has
    terminated
    //add CENTRE_AMOUNT (do not penalise again for use of 40) to
    //total for centre for all unsponsored members
    //return the correct string


function TMemberManager.spendingMoneyReport(filename : string):
     string;
  var
    output, sponsorName : string;
    i, centreSum, sponsorSum : integer;
    inFile : textfile;
  begin
    output := 'Spending Money Report' + #13#10;
    centreSum := 0;
    if FileExists(filename) <> true then
      begin
        WriteLn('File not found.');
      end
    else
      begin
        AssignFile(inFile, filename);
        Reset(inFile);
        while NOT EOF(inFile) do
          begin
            ReadLN(inFile, sponsorName);
            sponsorSum := 0;
            for i := 1 to size do
            begin
              if mArr[i] IS TSponsoredMember then
```

```
            begin
              if sponsorName = (mArr[i] AS
    TSponsoredMember).getSponsorName() then
              begin
                if (mArr[i] AS TSponsoredMember).getAmount() >=
    CENTRE_AMOUNT then
                begin
                  sponsorSum := sponsorSum + CENTRE_AMOUNT;
                end
                else
                begin
                  sponsorSum := sponsorSum + (mArr[i] AS
    TSponsoredMember).getAmount();
                  centreSum := centreSum + (CENTRE_AMOUNT -
    (mArr[i] AS TSponsoredMember).getAmount());
                end;

              end;

            end;
          end;
          output := output + sponsorName + ' total = $' +
    IntToStr(sponsorSum) + #13#10;
       end;

     end;
     for i := 1 to size do
       begin
         if NOT(mArr[i] IS TSponsoredMember) then
           centreSum := centreSum + CENTRE_AMOUNT;
       end;
     output := output + 'Centre Total = $' + IntToStr(centreSum) +
    #13#10;
     Result := output;

  end;

end.
```

QUESTION 5 & 6.3 & 7.2

```
//Question 5
//Question 5.1 - 1
//program created
program MemberUI;

{$APPTYPE CONSOLE}

{$R *.res}

uses
  System.SysUtils,
  DateUtils,
  Member in 'Member.pas',
  SponsoredMember in 'SponsoredMember.pas',
  MemberManager in 'MemberManager.pas';

var
  mm : TMemberManager;
//Question 5.2 - 1
//MemberManager created in correct position
begin
  try
    { TODO -oUser -cConsole Main : Insert code here }
    mm := TMemberManager.Create();
   //Question 5.3 - 1
   //print MemberManager object
   WriteLn(mm.toString);
   //Question 5.4 - 3
   //pass parameter "Stuart Redford"
   //call method ageOfMember correctly
   //method call inside WriteLn
   WriteLn(mm.ageOfMember('Stuart Redford'));

   //Question 6.3 - 1
   //call totalMemberTypes correctly, do not penalise for lack of /
        incorrect WriteLn
   WriteLn(mm.totalMemberTypes());

        //Question 7.2 - 1
        //call spendingMoneyReport correctly, do not penalise for lack of / incorrect
        WriteLn
        //correct parameters sent to spendingMoneyReport method
        WriteLn(mm.spendingMoneyReport('Sponsors.txt'));
```

Total: 150 marks