basic education

Department:
Basic Education
REPUBLIC OF SOUTH AFRICA

NATIONAL SENIOR CERTIFICATE

GRADE12

INFORMATION TECHNOLOGY P1

FEBRUARY/MARCH 2017

MEMORANDUM

MARKS: 150

This memorandum consists of 29 pages.

GENERAL INFORMATION:

- These marking guidelines are to be used as the basis for the marking session. They were prepared for use by markers. All markers are required to attend a rigorous standardisation meeting to ensure that the guidelines are consistently interpreted and applied in the marking of candidates' work.

- Note that learners who provide an alternate correct solution to that given as example of a solution in the marking guidelines will be given full credit for the relevant solution, unless the specific instructions in the paper were not followed or the requirements of the question were not met.

- **Annexures A, B and C** (pages 3–9) include the marking grid for each question for using either one of the two programming languages.

- **Annexures D, E and F** (pages 10–17) contain examples of solutions for Java for QUESTION 1 to QUESTION 3 in programming code.

- **Annexures G, H and I** (pages 18–29) contain examples of solutions for Delphi for QUESTION 1 to QUESTION 3 in programming code.

- Copies of **Annexures A, B and C** (pages 3-9) should be made for each learner and completed during the marking session.

ANNEXURE A:

SECTION A:

QUESTION 1: MARKING GRID- GENERAL PROGRAMMING SKILLS

CENTRE NUMBER:	**EXAMINATION NUMBER:**		
QUESTION	**DESCRIPTION**	**MAX. MARKS**	**LEARNER'S MARKS**
	A learner must be penalised only once if the same error is repeated.		
1.1	**Button - [Question 1_1]** Extract name and surname from textbox✓ Check if space is included in name surname string✓ If no space included✓ Display message indicating no space included ✓ Extract ID number✓ If ID number textbox is empty✓ Display message indicating the textbox is empty✓ If ID number string does not consist of 13 digits✓ Display message indicating 13 digits was not entered✓ If nameSurname string includes a space✓ AND ID number is not empty✓ AND ID number has 13 digits ✓ (Both conditions tested with AND)✓ Set Welcome message to visible✓	**14**	
1.2	**Buttons with Icons showing type of Loyalty card** **Button Discount:** Set global value to DI ✓ Change Discount heading to bold ✓ and Charity heading to not bold ✓ **Button Charity:** Set global value to CH ✓ Change Charity heading to bold ✓ and Discount heading to not bold ✓ **NOTE:** One mark for an attempt to bold the text on button One mark for an attempt to remove bold from the text on button	**6**	

1.3	**Button - [Question 1_3]** Extract purchase amount from text box ✓and convert to real value ✓ Use type of card code to test (Case/Switch or If): If 'DI' ✓ Calculate discount at 1.5% ✓ of amount ✓ Compile message ✓ amount displayed as currency ✓ with 2 decimal places ✓ If 'CH' ✓ Calculate number of R100's✓ * 3 ✓ Making provision for part of R100 ✓ Compile message ✓including the amount✓ Display message in text box ✓	**15**	
1.4	**Button - [Question 1_4]** **Part 1:** Assign a default character P to variable ✓ If Business card checked ✓ then assign B to variable ✓ Extract first 3 digits from ID number ✓and add to content of variable ✓ **Part 2:** Extract the last digit of the ID number ✓✓ Convert to integer ✓ Find the character at the integer position in given string ✓ Join the character to digit ✓ Display compiles strings in text boxes: Part 1 ✓ Part 2 ✓ Part 3 (DI/CH) ✓	**13**	
	TOTAL:	**48**	

ANNEXURE B:

SECTION B

QUESTION 2: MARKING GRID - OBJECT-ORIENTED PROGRAMMING

CENTRE NUMBER:	**EXAMINATION NUMBER:**		
QUESTION	**DESCRIPTION**	**MAX. MARKS**	**LEARNER'S MARKS**
2.1.1	**Constructor:** Definition with three correct parameters ✓ and correct data types✓ Assign parameter values to card number, cellphone number and loyalty points ✓ Initialise number of visits to 0 ✓ Initialise healthLevel attribute to S ✓	**5**	
2.1.2	**Mutator method:** setNumVisits method with parameter✓ no return ✓	**2**	
2.1.3	**increaseLoyaltyPoints method:** Method definition receiving total amount as parameter as real value✓ Determine loyalty points earned: Change total to integer (without rounding ✓ – full rands) ✓ Divide by 4✓ Add calculated points to loyalty points attribute✓	**5**	
2.1.4	**updateHealthLevel method:** Method definition receiving health amount and total amount as parameters✓ Calculate percentage ✓ If percentage is less than 10✓ Assign 'S' to healthLevel-attribute ✓ If percentage is 10 or more and less than 40✓ Assign 'G' to healthLevel-attribute ✓ If percentage is 40 or more Assign 'P" to healthLevel-attribute ✓	**7**	

2.1.5	**isCorrect method: (Method definition provided)** Loop through the digits in the ID number ✓ remove the 0 digits ✓ from the string Initialise variable for sum✓ If even number of digits left in string ✓ Loop correct number of times ✓ Add values with 2 digits to sum variable from left hand side of string ✓ If odd number of digits left in string ✓ Add first left hand side digit from string as a one digit value to sum✓ Loop correct number of times ✓ Add two-digits values to sum variable ✓ If sum is equal to access code parameter ✓ return true✓ else return false✓	**13**	
2.1.6	**identifyStarShopper method:** Empty string variable - if not Star shopper ✓ If (loyalty points > 2000✓ AND number of visits > 10)✓ OR✓ (healthLevel is 'P') ✓ Set variable to string 'STAR shopper'✓ Return string variable	**6**	

QUESTION 2: MARKING GRID – continued

2.2.1	**Button – [2.2.1 – Check access code]** Extract the card number from the combo box✓ Extract cellphone number from label ✓ Extract access code from text box ✓ Extract loyalty points from label and convert to integer✓ Instantiate object ✓ sending correct arguments in correct order✓ (6) **If access code is correct** Test if access code is correct ✓ using isCorrect method ✓ **Read text file and process data** {*Delphi: AssignFile, Reset and CloseFile* *Java: Create object to read from file*} ✓ Initialise variables for counters and sum ✓ Loop through file ✓ Read card number ✓ Read total amount spent and convert to double data type ✓ Read health amount spent and convert to double ✓ Test if it is correct card number ✓ Add total amount to sumTotal ✓ Add health amount to sumHealth ✓ Increment counter ✓ Call setVisits method ✓ Call increaseLoyaltyPoints method – send sumTotal as argument ✓ Call updateHealthLevel method – send sumTotal and sumHealth as arguments ✓ end loop Enable button for Q2.2.2 ✓ **If access code is NOT correct** Display message indicating that access code was incorrect ✓ Clear the access code text box ✓ (18)	**24**	
2.2.2	**Button – [2.2.2 - Display card holder details]** Display object details using toString ✓ in the output area ✓ Display shopper status using identifyStarShopper method ✓	**3**	
	TOTAL:	**65**	

ANNEXURE C:

SECTION C

QUESTION 3: MARKING GRID – PROBLEM SOLVING

CENTRE NUMBER:	**EXAMINATION NUMBER:**		
QUESTION	**DESCRIPTION**	**MAX. MARKS**	**LEARNER'S MARKS**
3.1	**Button [3.1 – Display layout]** Outer loop for rows ✓ Inner loop for columns ✓ Display symbol from two dimensional array ✓ In row ✓ and in column ✓	**5**	
3.2	**Combo box – Select Restaurants** Extract shop as character from combo box ✓ Initialise variables for position (row and column ✓) Loop through the rows ✓ If the shop is within characters A-E (to the left)✓ Set col for shop to 0 (first column) ✓ Set col for friend to 1 (second column) ✓ If the shop is within characters F-J (to the right) ✓ Set col for shop to last col of arrMall ✓ Set col for friend to one less than col for shop ✓ Test if character at current position in array ✓ equals selected shop's character ✓ Replace X with # ✓ Update display ✓ End loop	**13**	
3.3	**Button [3.3 - Locate nearest friend(s)]** Initialise nearest distance to a large value✓ Outer loop for rows ✓ Inner loop for columns/if for right hand side ✓ If character at position is # ✓ Determine difference in row positions ✓ as positive value ✓ Determine difference in column positions✓ as positive value ✓ Calculate newDistance ✓ If friend is in A-E or display shop name ✓ If friend is in F-J ✓ display shop name ✓ If newDistance < = nearest distance ✓ Add shop name to output message ✓ Replace nearest with newDistance ✓ Displayshop name ✓ and distance * 4.5 ✓ Display output message indicating nearest friends ✓	**19**	
	TOTAL:	**37**	

SUMMARY OF LEARNER'S MARKS:

CENTRE NUMBER:		**EXAMINATION NUMBER:**		
	SECTION A	**SECTION B**	**SECTION C**	
	QUESTION 1	**QUESTION 2**	**QUESTION 3**	**GRAND TOTAL**
MAX. MARKS	**48**	**65**	**37**	**150**
LEARNER'S MARKS				

ANNEXURE D: SOLUTION FOR QUESTION 1: JAVA

```
// Code provided
    String typeCard = "";
    String characters = "!@#$%]&*^~";

//************************************************************
// Question 1.1
// ************************************************************
private void btnQ1_1ActionPerformed(java.awt.event.ActionEvent evt)
{
  String nameSurname = txfNameSurname.getText();
  String idNum = txfIDNumber.getText();
  boolean nSpace = nameSurname.contains(" ");
  boolean noId = idNum.length() == 0;
  boolean idLen = idNum.length() != 13;
  if (!nSpace) {
      JOptionPane.showMessageDialog(rootPane, "The name-surname field
        does not contain a space.");
  }
  if (noId) {
      JOptionPane.showMessageDialog(rootPane, "An ID number must be
        entered.");
  }
  else
    if (idLen) {
        JOptionPane.showMessageDialog(rootPane, "The ID number does not
           consist of 13 digits.");
    }
  if (nSpace && !idLen && !noId) {
      lblWelcomeMessage.setVisible(true);
  }}
// ************************************************************
// Question 1.2
// ************************************************************
private void btn1_2DiscountActionPerformed
                     (java.awt.event.ActionEvent evt)
{
        typeCard = "DI";
        lblDiscount.setFont(new java.awt.Font("Tahoma", 1, 12));
        lblCharity.setFont(new java.awt.Font("Tahoma", 0, 12));
    }

    private void btn1_2CharityActionPerformed
                     (java.awt.event.ActionEvent evt) {
        typeCard = "CH";
        lblCharity.setFont(new java.awt.Font("Tahoma", 1, 12));
        lblDiscount.setFont(new java.awt.Font("Tahoma", 0, 12));
    }
```

```
// ************************************************************
// Question 1.3
// ************************************************************
private void btnQ1_3ActionPerformed(java.awt.event.ActionEvent evt) {
  double amount = Double.parseDouble(txfAmount.getText());
  String message = "";
  switch (typeCard) {
     case "DI":
           double discount = amount * 0.015;
           message = "1.5% discount on your purchase amount is " +
                   String.format("R %.2f", discount) + ".";
           break;
     case "CH":
           double contribution = Math.ceil(amount / 100.0) * 3;
           message = "An amount of " + String.format("R %.2f",
                   contribution) + " will be donated to charity.";
           break;
    }
   txfQ1_3.setText(message);    }
}

// ************************************************************
// Question 1.4
// ************************************************************
private void btnQ1_4ActionPerformed(java.awt.event.ActionEvent evt) {
    String bCard = "P";
   if (chkBusiness.isSelected())     {
     bCard = "B";
   }
   String part1 = bCard + txfIDNumber.getText().substring(0,3);
   char digit = txfIDNumber.getText().charAt(12);
   int pos = Integer.parseInt(digit + "");
   String part2 = digit + "" + characters.charAt(pos) + "";

   txfCode1.setText(part1 + "");
   txfCode2.setText(part2 + "");
   txfCode3.setText(typeCard);
}
```

ANNEXURE E: SOLUTION FOR QUESTION 2: JAVA

SOLUTION FOR QUESTION 2: OBJECT CLASS

```
package Question2Package;

public class CardHolder{
    // Code provided
    private String cardNumber;
    private String cellNumber;
    private int numVisits;
    private int loyaltyPoints;
    private char healthLevel;
//*****************************************************************
//Question 2.1.1
//*****************************************************************
public CardHolder(String cardNumber, String cellNumber, int
loyaltyPoints)
{
      this.cardNumber = cardNumber;
      this.cellNumber = cellNumber;
      this.loyaltyPoints = loyaltyPoints;
      this.numVisits = 0;
      this.healthLevel = 'S';
}
//*****************************************************************
//Question 2.1.2
//*****************************************************************
public void setNumVisits(int numVisits) {
        this.numVisits = numVisits;
    }
//*****************************************************************
//Question 2.1.3
//*****************************************************************
public void increaseLoyaltyPoints(double total) {
        loyaltyPoints = loyaltyPoints + (int) total / 4;
    }
//*****************************************************************
//Question 2.1.4
//*****************************************************************
public void updateHealthLevel(double sumHealth, double sumTotal) {
        double perc = sumHealth / sumTotal * 100;

        healthLevel = 'S';

        if (perc >= 10 && perc < 40) {
            healthLevel = 'G';
        }
        if (perc >= 40) {
            healthLevel = 'P';
        }
    }
```

```
//*********************************************************************
//Question 2.1.5
//*********************************************************************
public boolean isCorrect(int accessNum) {  //given
        String cNum = "";
        for (int cnt = 0; cnt < cellNumber.length(); cnt++) {
            if (cellNumber.charAt(cnt) != '0')
            {
                cNum = cNum + cellNumber.charAt(cnt);
            }
        }
        int sum = 0;
        if (cNum.length() % 2 == 0) {
           for (int cnt = 0; cnt < cNum.length(); cnt = cnt + 2) {
             sum = sum + Integer.parseInt(cNum.substring(cnt, cnt + 2));
            }
        } else {
            sum = Integer.parseInt(cNum.substring(0, 1));  //1
            for (int cnt = 1; cnt < cNum.length(); cnt = cnt + 2) {  //1
             sum = sum + Integer.parseInt(cNum.substring(cnt, cnt + 2));
          }
        }

        if (sum==accessNum) {
            return true;
        } else {
            return false;
        }
    }

//*********************************************************************
//Question 2.1.6
//*********************************************************************
public String identifyStarShopper() {
  String star = "";
  if ((loyaltyPoints > 2000 && numVisits > 10)||(healthLevel == 'P')) {
      star = "STAR shopper";
  }
  return star;
}
//*********************************************************************
// Code provided
//*********************************************************************
  public String toString() {
        return cardNumber + "\nContact number: " + cellNumber +
      "\n\nUpdated number of loyalty points: " + loyaltyPoints +
      "\nNumber of visits: " + numVisits + "\nHealth evaluation status:
      " + healthLevel;
    }
}
```

GUI CLASS: QUESTION2_SOLUTION

```
//*********************************************************************
// Code provided
//*********************************************************************

    CardHolder objCardHolder = null;

    public Question2_Memo() {
        initComponents();
        setLocationRelativeTo(this);
        btnQues222.setEnabled(false);
    }

//*********************************************************************
//Question 2.2.1
//*********************************************************************
private void btnQ221ActionPerformed(java.awt.event.ActionEvent evt) {
  String cardNum = "" + cmbCardNumbers.getSelectedItem();
  String cellNum = lblCellNumber.getText();
  int accessNum = Integer.parseInt(txfCode.getText());
  int currPoints = Integer.parseInt(lblLoyaltyPoints.getText());
  objCardHolder = new CardHolder(cardNum, cellNum, currPoints);
  if (objCardHolder.isCorrect(accessNum)) {
      JOptionPane.showMessageDialog(rootPane, "The access code is
                                                    correct.");
   try {
     Scanner scFile = new Scanner(new
                      FileReader("DataJanuary2017.txt"));
     double sumTotal = 0;
     double sumHealth = 0;
     int cnt = 0;
     while (scFile.hasNext()) {
       String card = scFile.nextLine();
       double totalAmount =
                           Double.parseDouble(scFile.nextLine());
       double healthAmount =
                           Double.parseDouble(scFile.nextLine());
       if (card.equalsIgnoreCase(cardNum)) {
         sumTotal = sumTotal + totalAmount;
         sumHealth = sumHealth + healthAmount;
         cnt++;
       } // if
     } // while
     objCardHolder.setNumVisits(cnt);
     objCardHolder.increaseLoyaltyPoints(sumTotal);
     objCardHolder.updateHealthLevel(sumHealth, sumTotal);
    } catch (FileNotFoundException e) {
       JOptionPane.showMessageDialog(rootPane, e);
    }
    btnQ222.setEnabled(true);
   } else {
       JOptionPane.showMessageDialog(rootPane, "Incorrect access
                                                    code.");
  txfCode.setText("");
 }
```

```
//*********************************************************************
//Question 2.2.2
//*********************************************************************
private void btnQues222ActionPerformed(java.awt.event.ActionEvent evt)
{
        txaOutput.setText(objCardHolder.toString());
        txaOutput.append("\n" + objCardHolder.identifyStarShopper());
}
//*********************************************************************
// Code provided
//*********************************************************************
public void updatePoints() {
        if (cmbCardNumbers.getSelectedIndex() == 0) {
            lblLoyaltyPoints.setText("2130");
            lblCellNumber.setText("0812345678");
        }
        if (cmbCardNumbers.getSelectedIndex() == 1) {
            lblLoyaltyPoints.setText("5723");
            lblCellNumber.setText("0822001100");
        }
        if (cmbCardNumbers.getSelectedIndex() == 2) {
            lblLoyaltyPoints.setText("12908");
            lblCellNumber.setText("0740998877");
        }
        if (cmbCardNumbers.getSelectedIndex() == 3) {
            lblLoyaltyPoints.setText("500");
            lblCellNumber.setText("0720951083");
        }
        txaOutput.setText("");
        txfCode.setText("");
        btnQ222.setEnabled(false);
    }
```

ANNEXURE F: SOLUTION FOR QUESTION 3: JAVA

```
//*********************************************************************
// Code provided
//*********************************************************************
char[][] arrMall = {{'A', 'X', 'O', '*', 'O', 'X', 'J'}, {'B', 'X', 'O',
'O', 'O', 'X', 'I'}, {'C', 'X', 'O', 'O', 'O', 'X', 'H'}, {'D', 'X',
'O', 'O', 'O', 'X', 'G'}, {'E', 'X', 'O', 'O', 'O', 'X', 'F'}};
//*********************************************************************
// Question 3.1
//*********************************************************************
private void btnDisplayActionPerformed(java.awt.event.ActionEvent evt) {
        txaQ31.setText("\n");
        for (int r = 0; r < arrMall.length; r++) {
            for (int c = 0; c < arrMall[0].length; c++) {
                txaQ31.append(String.format("%5s", arrMall[r][c]));
            }
            txaQ31.append("\n");
        }
    }

//*********************************************************************
// Question 3.2
//*********************************************************************
    private void cmbShopsActionPerformed(java.awt.event.ActionEvent evt)
{
        char shop = ((String) cmbShops.getSelectedItem()).charAt(0);
        int colShop = 0;
        int colFriend = 0;
        for (int r = 0; r < arrMall.length; r++) {
            if (shop >= 'A' && shop <= 'E') {
                colShop = 0;
                colFriend = 1;
            } else {
                colShop = arrMall[0].length - 1;
                colFriend = colShop - 1;
            }
            if (arrMall[r][colShop] == shop) {
                arrMall[r][colFriend] = '#';
            }
        }
        btnDisplay.doClick();
    }
```

```
//*********************************************************************
// Question 3.3
//*********************************************************************

private void btnLocateNearActionPerformed(java.awt.event.ActionEvent
evt) {
        int dist = 0;
        int nDist = 100;
        String messSentence = "Your nearest friend(s) is/are at:  ";
        String mess = "";
        char rest = ' ';
        txaQ33.setText("");
        for (int r = 0; r < arrMall.length; r++) {
            for (int c = 0; c < arrMall[0].length; c++) {
                if (arrMall[r][c] == '#') {
                    int diffR = Math.abs(r - rYou);
                    int diffC = Math.abs(c - cYou);
                    dist = diffR + diffC +1;
                    if (c == 5) {
                        rest = arrMall[r][c + 1];
                    } else {
                        rest = arrMall[r][c - 1];
                    }
                    txaQ33.append("Restaurant " + rest + " -
                      approximately " + (dist*4.5) + " metres\n");
                    if (dist <= nDist) {
                        nDist = dist;
                        mess = mess + " " + rest;
                   }
                }
            }
        }
        txaQ33.append("\n" + messSentence + mess);
    }
```

ANNEXURE G: SOLUTION FOR QUESTION 1: DELPHI

```
unit Question1_U;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
Forms, Dialogs, StdCtrls, ExtCtrls, pngimage, Math;

type
  TfrmQuestion1 = class(TForm)
    pnlHeading: TPanel;
    gpbQuestion11: TGroupBox;
    Label1: TLabel;
    Label2: TLabel;
    edtName: TEdit;
    edtIDNumber: TEdit;
    Label3: TLabel;
    btnQuest11: TButton;
    gpbQuestion12: TGroupBox;
    Label4: TLabel;
    lblDiscount: TLabel;
    lblCharity: TLabel;
    imgDiscount: TImage;
    imgCharity: TImage;
    gpbQuest13: TGroupBox;
    Label7: TLabel;
    edtAmount: TEdit;
    edtDisplay: TEdit;
    btnQuest13: TButton;
    gpbQuestion14: TGroupBox;
    ckbBusiness: TCheckBox;
    Label9: TLabel;
    edtPart1: TEdit;
    Label10: TLabel;
    edtPart2: TEdit;
    Label11: TLabel;
    edtPart3: TEdit;
    btnQuest14: TButton;
    lblWelcome: TLabel;
    Label5: TLabel;
    procedure btnQuest11Click(Sender: TObject);
    procedure imgDiscountClick(Sender: TObject);
    procedure imgCharityClick(Sender: TObject);
    procedure btnQuest13Click(Sender: TObject);
    procedure btnQuest14Click(Sender: TObject);
    procedure FormActivate(Sender: TObject);
private
    { Private declarations }
  public
    { Public declarations }
  end;
```

```
var

  sName,sIdNumber: string;

// ****************************************************************
// Code provided
// ****************************************************************
  frmQuestion1: TfrmQuestion1;
  typeCard: string;
  characters: string = '!@#$%]&*^~';

implementation

{$R *.dfm}

// ****************************************************************
// Question 1.1
// ****************************************************************
procedure TfrmQuestion1.btnQuest11Click(Sender: TObject);
var
    bNoSpace, bNoID, bIDLen: boolean;
begin
  sName := edtName.Text;
  bNoSpace := pos(' ',sName) = 0;
  if bNoSpace then
      showMessage('The name-surname field does not contain a space.');
  sIdNumber := edtIDNumber.Text;
  bNoID := length(sIDNumber)= 0;
  bIDLen := length(sIDNumber) <> 13;
  if bNoID then
     showMessage('An ID number must be entered.')
  else if bIDLen then
     showMessage('The ID number does not consist of 13 digits.');

  if (bNoSpace = false) AND (bNoID = false) AND (bIDLen = false) then
      lblWelcome.visible := true;
end;

// ****************************************************************
// Question 1.2
// ****************************************************************
procedure TfrmQuestion1.imgDonationClick(Sender: TObject);
begin
  lblCharity.Font.Name := 'Arial';
  lblCharity.Font.Style := [fsBold];
  lblDiscount.Font.Style := [];
  typeCard:='CH';
end;

procedure TfrmQuestion1.imgDiscountClick(Sender: TObject);
begin
  lblDiscount.Font.Name := 'Arial';
  lblDiscount.Font.Style := [fsBold];
  lblCharity.Font.Style := [];
  typeCard := 'DI';
end;
```

```
// ************************************************************
// Question 1.3
// ************************************************************
procedure TfrmQuestion1.btnQuest13Click(Sender: TObject);
var
  rDiscount,rDonation: real;
begin
  if typeCard = 'DI' then
  begin
    rDiscount := 1.5/100 * StrToFloat(edtAmount.Text);
    edtDisplay.Text := '1.5% discount on your purchase amount is ' +
           FloatToStrF(rDiscount,ffCurrency,6,2) + '.';
  end
  else
  begin
    rDonation := Math.Ceil(StrToFloat(edtAmount.Text) / 100)*3;
    edtDisplay.Text := 'An amount of '+FloatToStrF
           (rDonation,ffCurrency,6,2)+' will be donated to charity.';
  end
end;
// ************************************************************
// Question 1.4
// ************************************************************
procedure TfrmQuestion1.btnQuest14Click(Sender: TObject);
var
  sSymbol  : string;
  iPosition: integer;
  bCard    : char;
begin
  if ckbBusiness.Checked then
    bCard:='B'
  else
    bCard:='P';
  iPosition := StrToInt(sIdNumber[length(sIdNumber)]);
  sSymbol := characters[iPosition+1];
  edtPart1.Text := bCard + copy(sIDNumber,1,3);
  edtPart2.Text := IntToStr(iPosition)+sSymbol;
  edtPart3.Text := typeCard;
end;

// ************************************************************
// Code provided
// ************************************************************
procedure TfrmQuestion1.FormActivate(Sender: TObject);
begin
  CurrencyString := 'R';
end;

end.
```

ANNEXURE H: SOLUTION FOR QUESTION 2: DELPHI

OBJECT CLASS:

```
unit CardHolder_U;

interface
  uses SysUtils,Math;
type
  TCardHolder = class(TObject)
  private
    fCardNumber: string;
    fCellNumber: string;
    fNumVisits: integer;
    fLoyaltyPoints: integer;
    fHealthLevel: char;
  public
  constructor create(sCardNumber,sCellNumber: string; iLoyaltyPoints:
integer);
  procedure setNumVisits(iNumVisits: integer);
  Procedure increaseLoyaltyPoints(rTotal: real);
  procedure updateHealthLevel(rHealth,rTotal: real);
  function isCorrect(sAccessCode:string): boolean;
  function identifyStarShopper: string;
  function toString: string;
  end;

implementation

{ TCardHolder }

// ************************************************************
// Question 2.1.1
// ************************************************************
constructor TCardHolder.create(sCardNumber, sCellNumber: string;
                iLoyaltyPoints: integer);
begin
    fCardNumber := sCardNumber;
    fCellNumber := sCellnumber;
    fLoyaltyPoints := iLoyaltyPoints;
    fNumVisits := 0;
    fHealthLevel := 'S';
end;

// ************************************************************
// Question 2.1.2
// ************************************************************
procedure TCardHolder.setNumVisits(iNumVisits: integer);
begin
  fNumVisits:= iNumVisits;
end;
```

```
// ************************************************************
// Question 2.1.3
// ************************************************************
procedure TCardHolder.increaseLoyaltyPoints(rTotal: real);
var
  iLoyaltyPoints: integer;
begin
  fLoyaltyPoints := fLoyaltyPoints + trunc(rTotal) div 4;
end;

// ************************************************************
// Question 2.1.4
// ************************************************************
procedure TCardHolder.updateHealthLevel(rHealth, rTotal: real);
var
  rPercent: real;
begin
  rPercent := rHealth/rTotal * 100;
  if rPercent < 10  then
    fHealthLevel:='S'
  else
    if rPercent >= 40 then
       fHealthLevel := 'P'
    else
       fHealthLevel := 'G'
end;

// ************************************************************
// Question 2.1.5
// ************************************************************
// Provided definition

function TCardHolder.isCorrect(sAccessCode: string): boolean;
var
  a: integer;
  b,iSum: integer;
  sCellNumber : string;
begin
  iSum:=0;
  sCellNumber := fCellNumber;
  for a := length(sCellNumber) downto 1 do
    if sCellNumber[a] = '0' then
      delete(sCellNumber,a,1);
  if length(sCellNumber) mod 2 = 0  then
    for b := 1 to (length(sCellNumber) div 2) do
      iSum := iSum + StrToInt(copy(sCellNumber,2*b-1,2));
  if length(sCellNumber) mod 2 <> 0  then
   begin
     iSum:=StrToInt(sCellNumber[1]);
     for b := 1 to (length(sCellNumber) div 2) do
      iSum := iSum + StrToInt(copy(sCellNumber,2*b,2));
   end;
  if StrToInt(sAccessCode) = iSum then
    result:=true
  else
    result := false;
end;
```

```
// ************************************************************
// Question 2.1.6
// ************************************************************
function TCardHolder.identifyStarShopper: string;
begin
  result := '';
  if ((fLoyaltyPoints > 2000) AND ( fnumVisits > 10)) OR ( fHealthLevel
= 'P') then
     result := 'STAR shopper'
end;

// ************************************************************
// Code provided
// ************************************************************
function TCardHolder.toString: string;
begin
  result:= fCardNumber + #13 + 'Contact number: '+ fCellNumber +
           13+#13+ 'Updated number of loyalty points:
           +IntToStr(fLoyaltyPoints)+#13+'Number of visits:
           '+IntToStr(fNumVisits)+#13+'Health evaluation status:
           '+fHealthLevel;
end;

end.
```

MAIN FORM UNIT: QUESTION2_U.PAS

```
unit Question2_U;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
Forms, Dialogs, StdCtrls, ExtCtrls, ComCtrls, CardHolder_U;

type
  TfrmQuestion2 = class(TForm)
    Panel1: TPanel;
    gbpQuestion2_1: TGroupBox;
    gpbQuestion2_2: TGroupBox;
    Label1: TLabel;
    lblCardNumber: TLabel;
    cmbCardNumbers: TComboBox;
    Panel2: TPanel;
    Label2: TLabel;
    lblLoyaltyPoints: TLabel;
    lblCellNumber: TLabel;
    Panel3: TPanel;
    Label3: TLabel;
    edtCode: TEdit;
    btnQuest221: TButton;
    redOutput: TRichEdit;
    btnQuest222: TButton;
    procedure cmbCardNumbersChange(Sender: TObject);
    procedure btnQuest221Click(Sender: TObject);
    procedure btnQuest222Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  frmQuestion2: TfrmQuestion2;
  objCardholder : TCardHolder;
  myFile:Textfile;

implementation

{$R *.dfm}

// ************************************************************
// Question 2.2.1
// ************************************************************
procedure TfrmQuestion2.btnQuest221Click(Sender: TObject);
var
  sCardHolder: string;
  sCellNumber: string;
  sLoyaltyPoints: string;
  sAccess: string;
```

```
  sLineOne, sLineTwo, sLineThree: string;

  rTotalSpent, rTotalHealth: real;
  iVisits: integer;
begin
  sCardHolder := cmbCardNumbers.items[cmbCardNumbers.ItemIndex];
  sCellNumber := lblCellNumber.Caption;
  sLoyaltyPoints := lblLoyaltyPoints.Caption;

  objCardholder := TCardHolder.create(sCardHolder, sCellNumber,
    StrToInt(sLoyaltyPoints));
  sAccess := edtCode.Text;
  if objCardholder.isCorrect(sAccess) then
  begin
    showMessage('The access code is correct.');
    rTotalSpent := 0;
    rTotalHealth := 0;
    iVisits := 0;
    assignFile(myFile, 'DataJanuary2017.txt');
    reset(myFile);
    while not eof(myFile) do
    begin
      readln(myFile, sLineOne);
      readln(myFile, sLineTwo);
      readln(myFile, sLineThree);
      if sLineOne = sCardHolder then
      begin
        rTotalSpent := rTotalSpent + StrToFloat(sLineTwo);
        rTotalHealth := rTotalHealth + StrToFloat(sLineThree);
        inc(iVisits);
      end; // end if
    end; // end while
    closeFile(myFile);

    objCardholder.increaseLoyaltyPoints(rTotalSpent);
    objCardholder.setNumVisits(iVisits);
    objCardholder.updateHealthLevel(rTotalHealth, rTotalSpent);
    btnQuest222.Enabled := true;
  end // end isCorrect
  else
  begin
    showMessage('Incorrect access code.');
    edtCode.Clear;
  end;
end;

// ************************************************************
// Question 2.2.2
// ************************************************************

procedure TfrmQuestion2.btnQuest222Click(Sender: TObject);
begin
  redOutput.Clear;
  redOutput.Lines.Add(objCardholder.toString);
  redOutput.Lines.Add(objCardholder.identifyStarShopper);
end;
```

```
// ************************************************************
// Code provided
// ************************************************************

procedure TfrmQuestion2.cmbCardNumbersChange(Sender: TObject);
begin
  edtCode.Clear;
  redOutput.Clear;
  btnQuest222.Enabled := false;
  case cmbCardNumbers.ItemIndex of
    0:
      begin
        lblLoyaltyPoints.Caption := '2130';
        lblCellNumber.Caption := '0812345678';
      end;

    1:
      begin
        lblLoyaltyPoints.Caption := '5723';
        lblCellNumber.Caption := '0822001100';
      end;

    2:
      begin
        lblLoyaltyPoints.Caption := '12908';
        lblCellNumber.Caption := '0740998877';
      end;

    3:
      begin
        lblLoyaltyPoints.Caption := '500';
        lblCellNumber.Caption := '0720951083';
      end;
  end;

end.
```

ANNEXURE I: SOLUTION FOR QUESTION 3: DELPHI

```
unit Question3_U;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
Forms, Dialogs, StdCtrls, ComCtrls, ExtCtrls, Grids;

type
  TfrmQuestion3 = class(TForm)
    GroupBox1: TGroupBox;
    Panel1: TPanel;
    btnQuest31: TButton;
    btnQuest33: TButton;
    cmbShops: TComboBox;
    GroupBox2: TGroupBox;
    redOutput: TRichEdit;
    Label1: TLabel;
    redGrid: TRichEdit;
    procedure btnQuest31Click(Sender: TObject);
    procedure cmbShopsChange(Sender: TObject);
    procedure btnQuest33Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  frmQuestion3: TfrmQuestion3;

//*********************************************************************
// Provided code
//*********************************************************************

  arrMall:array[1..5,1..7] of string
=(('A','X','O','*','O','X','J'),('B','X','O','O','O','X','I'),('C','X','
O','O','O','X','H'),('D','X','O','O','O','X','G'),('E','X','O','O','O','
X','F'));

  arrPlace: array[1..10] of string;
implementation
{$R *.dfm}
```

```
// ************************************************************
// Question 3.1
// ************************************************************
procedure TfrmQuestion3.btnQuest31Click(Sender: TObject);
var
  iRow, iCol: integer;
  sRow      : string;
begin
  redGrid.Lines.clear;
  redGrid.Lines.add(' ');

  for iRow := 1 to 5 do
   begin
   sRow := '   ';
    for iCol := 1 to 7 do
      sRow := sRow + arrMall[iRow,iCol] + #9;
    redGrid.Lines.add(sRow);
    redGrid.Lines.add(' ');
   end;
end;

// ************************************************************
// Question 3.2
// ************************************************************
procedure TfrmQuestion3.cmbShopsChange(Sender: TObject);
var
  iRowChange:integer;
begin
  if cmbShops.Items[cmbShops.ItemIndex][1] in ['A'..'E'] then
    begin
     iRowChange:=cmbShops.ItemIndex + 1;
     arrMall[iRowchange,2]:='#'
    end
  else
    begin
     iRowChange:=10-cmbShops.ItemIndex ;
     arrMall[iRowchange,6]:='#';
    end;
  btnQuest31.Click;
end;
```

```
// ************************************************************
// Question 3.3
// ************************************************************
procedure TfrmQuestion3.btnQuest33Click(Sender: TObject);
var
  iRow,a: integer;
  iCol,iCounter: integer;
  rDistance,rLowest: real;
  sPlace:string;
begin
  redOutput.Clear;
  rLowest:=100;
  iCounter:=1;
  sPlace:='';
  for iRow := 1 to 5 do
   begin
    if arrMall[iRow,2] = '#' then
       begin
         rDistance:= 4.5*(iRow+1)+4.5;
         if rDistance <= rLowest then
            begin
             inc(iCounter);
             rLowest:=rDistance;
              arrPlace[iCounter]:=arrMall[iRow,1];
            end;
         redOutput.Lines.Add('Restaurant '+ arrMall[iRow,1]+' -
           approximately '+FloatToStrF(rDistance,ffFixed,4,1)+ '
           metres');
       end;
       if arrMall[iRow,6] = '#' then
       begin
         rDistance:= 4.5*(iRow+1)+4.5;
         if rDistance <= rLowest then
            begin
             inc(iCounter);
             rLowest:=rDistance;
              arrPlace[iCounter]:=arrMall[iRow,7];
            end;
         redOutput.Lines.Add('Restaurant '+ arrMall[iRow,7]+' -
           approximately '+FloatToStrF(rDistance,ffFixed,4,1)+ '
           metres');
       end;
   end;
     for a := 1 to iCounter do
       sPlace := sPlace + ' '  +arrPlace[a];

 redOutput.Lines.Add(' ');
 redOutput.Lines.Add('Your nearest friend(s) is/are at: '+ sPlace);
end;

end.
```