**THE UNITED REPUBLIC OF TANZANIA**

**NATIONAL EXAMINATIONS COUNCIL OF TANZANIA**

**ADVANCED CERTIFICATE OF SECONDARY EDUCATION EXAMINATION**

**136/2** **COMPUTER SCIENCE 2**

(For Both School and Private Candidates)

**Time: 3 Hours** **ANSWERS** **Year: 2013**

**Instructions:**

1. this paper consists of three questions.
2. Answer two questions including question number one
3. Submit printed codes and screenshots together with the softcopy of your work(s)

1. (a) Write a C++ program that reads an integer value from the user. A program should check and display a message whether the entered integer value is a perfect number or not.
(HINT: A perfect number is one whose sum of its factors excluding the number itself is equal to a number itself. E.g., 6 is a perfect number as the sum of its factors 1 + 2 + 3 = 6). Test the program and provide the screenshot of the test results.

Solution:

```
#include <iostream>
using namespace std;

int main() {
   int number, sum = 0;

   cout << "Enter an integer: ";
   cin >> number;

   for (int i = 1; i <= number / 2; i++) {
      if (number % i == 0) {
         sum += i;
      }
   }

   if (sum == number && number != 0) {
      cout << number << " is a perfect number." << endl;
   } else {
      cout << number << " is not a perfect number." << endl;
   }

   return 0;
}
```

This program calculates the sum of all divisors of a number (excluding the number itself) and compares the sum to the original number. If the sum equals the number, it is identified as a perfect number. For example, for an input of 6, the output will confirm it as a perfect number.

(b) Write a switch-case C++ program which can help users to convert temperature values from Fahrenheit to Centigrade and vice versa by entering the following options:
First option: Changes Fahrenheit to Centigrade.
Second option: Changes Centigrade to Fahrenheit.
The program should prompt the user to enter the value to be converted. Test the program and provide the screenshot of the test results.

Note: Use the following formula in developing your program:
(i) From Fahrenheit (F) to Centigrade (C), use C = (5.0/9.0) × (F - 32).
(ii) From Centigrade (C) to Fahrenheit (F), use F = (9.0/5.0) × C + 32.

Solution:

```
#include <iostream>
using namespace std;

int main() {
   int option;
   double temperature, convertedTemp;

   cout << "Temperature Conversion Program" << endl;
   cout << "1. Convert Fahrenheit to Centigrade" << endl;
   cout << "2. Convert Centigrade to Fahrenheit" << endl;
   cout << "Enter your choice (1 or 2): ";
   cin >> option;

   switch (option) {
      case 1:
         cout << "Enter temperature in Fahrenheit: ";
         cin >> temperature;
         convertedTemp = (5.0 / 9.0) * (temperature - 32);
         cout << "Temperature in Centigrade: " << convertedTemp << "°C" << endl;
         break;
      case 2:
         cout << "Enter temperature in Centigrade: ";
         cin >> temperature;
         convertedTemp = (9.0 / 5.0) * temperature + 32;
         cout << "Temperature in Fahrenheit: " << convertedTemp << "°F" << endl;
         break;
      default:
         cout << "Invalid option. Please select 1 or 2." << endl;
         break;
   }

   return 0;
}
```

This program uses a switch-case structure to determine whether to convert from Fahrenheit to Centigrade or Centigrade to Fahrenheit. The user is prompted to input a temperature and the conversion result is displayed. For example:
- Input: Option 1, Temperature 100°F

- Output: Temperature in Centigrade: 37.7778°C

2. (a) (i) Design a simple calculator using Microsoft Visual Basic program as shown in the snapshot below:

Write Visual Basic codes (program) which should ask the user to enter the first number through a message box after clicking any operator and prompts a user to enter the second number after clicking the ok button in the message box. The program should display the answer in the text box after clicking a command button ok, which appears in the message box that asks a user to enter the second number.

Perform all necessary formatting as directed below:
- All command buttons should be of equal size.
- Vertical and horizontal spacing of command buttons should be equal.
- The width and height of the command button should be 500 and 350, respectively.
- Set #H80000018 or Tool/Tip as the background color of the form.
- Substitute S.xxx/yyy, which appeared in the simple calculator interface, with your Examination Number.
- Write Visual Basic codes in such a way that when a user clicks any number, the number clicked should appear in the text box.

Solution:

```
' Simple Calculator Program in Visual Basic
Private Sub cmdAdd_Click()
   Dim num1 As Double, num2 As Double
   num1 = Val(InputBox("Enter the first number:"))
   num2 = Val(InputBox("Enter the second number:"))
   txtResult.Text = num1 + num2
End Sub

Private Sub cmdSubtract_Click()
   Dim num1 As Double, num2 As Double
   num1 = Val(InputBox("Enter the first number:"))
   num2 = Val(InputBox("Enter the second number:"))
   txtResult.Text = num1 - num2
End Sub

Private Sub cmdMultiply_Click()
   Dim num1 As Double, num2 As Double
   num1 = Val(InputBox("Enter the first number:"))
   num2 = Val(InputBox("Enter the second number:"))
   txtResult.Text = num1 * num2
End Sub

Private Sub cmdDivide_Click()
   Dim num1 As Double, num2 As Double
```

```
    num1 = Val(InputBox("Enter the first number:"))
    num2 = Val(InputBox("Enter the second number:"))
    If num2 <> 0 Then
        txtResult.Text = num1 / num2
    Else
        txtResult.Text = "Error: Division by zero"
    End If
End Sub

Private Sub cmdClear_Click()
    txtResult.Text = ""
End Sub

Private Sub cmdExit_Click()
    End
End Sub
```

Form Design Guidelines:
1. Create a form with six buttons labeled "+", "-", "*", "/", "Clear", and "Exit".
2. Set the size of each button to 500x350 pixels.
3. Add a text box named `txtResult` to display the result.
4. Set the form's background color to `#H80000018`.
5. Ensure the buttons are evenly spaced both horizontally and vertically.
When any operator button is clicked, the program asks the user to enter two numbers and performs the corresponding operation. The result is displayed in the `txtResult` text box.

2. (b) (i) Design a Visual Basic page (user interface) which consists of two labels, a textbox, a ListBox with a vertical scroll bar, and three command buttons. Name the controls as directed below:

Control | Name
Label1 | Enter Number
Label2 | List of Factors
Command1 | Find Factors
Command2 | Reset
Command3 | Cancel

Align labels in the first column, text, and ListBox in the second column, and Command buttons in the third column.

Scroll descriptions:
- The maximum scroll is 200.
- The minimum scroll is 100.

(ii) Write Visual Basic codes which will help a user to find factors of a number entered in the textbox. The factors should be displayed in the ListBox, which contains a scroll bar. When a user clicks a reset command button, the number entered by a user should be 0, and the interface should disappear when a user clicks a cancel button.

Solution:

```
' Visual Basic Code for the Factor Finder Program

Private Sub cmdFindFactors_Click()
   Dim num As Integer
   Dim i As Integer

   ListBox1.Clear
   num = Val(txtNumber.Text)

   If num <= 0 Then
      MsgBox "Please enter a positive number.", vbExclamation
      Exit Sub
   End If

   For i = 1 To num
      If num Mod i = 0 Then
         ListBox1.AddItem i
      End If
   Next i
End Sub

Private Sub cmdReset_Click()
   txtNumber.Text = "0"
   ListBox1.Clear
End Sub

Private Sub cmdCancel_Click()
   Unload Me
End Sub
```

Design Guidelines for the Form:
1. Add two labels:
   - Label1: Caption as "Enter Number"
   - Label2: Caption as "List of Factors"

2. Add one TextBox:
  - Name: `txtNumber`
  - Default value: "0"

3. Add one ListBox:
  - Name: `ListBox1`
  - Enable the vertical scroll bar by setting the `ScrollBars` property to `Vertical`.

4. Add three buttons:
  - Button1: Name: `cmdFindFactors`, Caption: "Find Factors"
  - Button2: Name: `cmdReset`, Caption: "Reset"
  - Button3: Name: `cmdCancel`, Caption: "Cancel"

5. Align the labels in the first column, the TextBox and ListBox in the second column, and the buttons in the third column.

This program calculates the factors of the entered number and displays them in the ListBox. The Reset button clears the input and resets the ListBox, while the Cancel button closes the form.

3. (a) Use HTML codes to create a page below. Set #F5DEB3 as the background color and activate a search textbox so that a user can search student ID/student name through Google search engine.

Student Details
Student Id:
Student Name:
Gender: ○ Male ○ Female
Subject: [Dropdown menu with options Science, Art, and Business]
[Submit button]

Note:  Use a table aligned at the center with border 0 to create a page in 3(a).
Solution:

```
<!DOCTYPE html>
<html>
<head>
   <title>Student Details</title>
   <style>
      body {
         background-color: #F5DEB3;
         text-align: center;
         font-family: Arial, sans-serif;
      }
      table {
```

```
        margin: 0 auto;
        border: 0;
      }
    </style>
</head>
<body>
    <h1>Student Details</h1>
    <form action="https://www.google.com/search" method="get" target="_blank">
      <table>
        <tr>
          <td>Student ID:</td>
          <td><input type="text" name="q" placeholder="Enter Student ID"></td>
        </tr>
        <tr>
          <td>Student Name:</td>
          <td><input type="text" name="q" placeholder="Enter Student Name"></td>
        </tr>
        <tr>
          <td>Gender:</td>
          <td>
            <input type="radio" name="gender" value="Male"> Male
            <input type="radio" name="gender" value="Female"> Female
          </td>
        </tr>
        <tr>
          <td>Subject:</td>
          <td>
            <select name="subject">
              <option value="Science">Science</option>
              <option value="Art">Art</option>
              <option value="Business">Business</option>
            </select>
          </td>
        </tr>
        <tr>
          <td colspan="2" style="text-align: center;">
            <button type="submit">Search</button>
          </td>
        </tr>
      </table>
    </form>
</body>
</html>
```

This code sets the background color to `#F5DEB3` and creates a table-aligned form with a search feature that uses Google. The search textbox is activated to allow queries for student ID or name.

3. (b) Alpha Training College offers three courses. A student sits for three exams every semester, each exam marked out of 100. The following is sample data collected from the college database.

Exams offered:

| Examination Code | Examination Name | Course Code | Exams Record No |
|---|---|---|---|
| 01 | Opener | D-SECT | 1 |
| 02 | Midterm | D-INT | 2 |
| 03 | Endterm | D-ACCT | 3 |

Exams performance:

| Exam Record No | Student ID | Exam Semester | Score |
|---|---|---|---|
| 1 | SECT-01 | 2 | 75 |
| 3 | ACCT-04 | 1 | 65 |
| 6 | INT-03 | 1 | 68 |
| 2 | INT-03 | 2 | 80 |
| 4 | SECT-01 | 2 | 70 |
| 5 | ACCT-04 | 2 | 60 |
| 7 | SECT-01 | 3 | 78 |
| 8 | INT-03 | 3 | 74 |
| 9 | ACCT-04 | 3 | 66 |

Courses:

| Course Code | Course Description | Tuition Fees |
|---|---|---|
| D-SECT | Dip. in Secretarial | Tsh. 150000 |
| D-INT | Dip. in IT | Tsh. 200000 |
| D-ACCT | Dip. in Accounting | Tsh. 160000 |

Tasks:

(i) Create a database file called Alpha College and save it.

(ii) Create a table structure for each of the three tables, setting the most appropriate field as the primary key and choosing the appropriate data type for each field.

(iii) Relate three tables as required to have one-to-many relationships.

Solution:

(i) Creating the Database:

Create a database file called Alpha_College.accdb using Microsoft Access or any database software of choice.

(ii) Table Structures:

1. Table: Exams_Offered

| Field Name | Data Type | Description |
|---|---|---|
| Exam_Code | Short Text | Primary Key |
| Exam_Name | Short Text | Name of the examination |
| Course_Code | Short Text | Foreign Key |
| Exam_Record_No | Number | Examination record ID |

2. Table: Exams_Performance

| Field Name | Data Type | Description |
|---|---|---|
| Exam_Record_No | Number | Foreign Key (Primary Key) |
| Student_ID | Short Text | Student Identifier |
| Exam_Semester | Number | Semester number |
| Score | Number | Marks scored |

3. Table: Courses

| Field Name | Data Type | Description |
|---|---|---|
| Course_Code | Short Text | Primary Key |
| Course_Description | Short Text | Name of the course |
| Tuition_Fees | Currency | Course fee amount |

(iii) Relationships:

1. Relationship 1: Courses.Course_Code → Exams_Offered.Course_Code (One-to-Many)
2. Relationship 2: Exams_Offered.Exam_Record_No → Exams_Performance.Exam_Record_No (One-to-Many)

These relationships establish links between courses, exams offered, and student performance.