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: 2015**

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) Create a C++ program that will prompt the user to enter student's final grades. Grades must be integer values ranging from 1 to 9. The program should prompt a user to key "Y or y" in order to enter more grades by displaying the statement "Do you want to enter more Grades Y/N?". The program then displays the grades average, the highest grade, and the number of students failed the course once a user keys "N or n". (HINT: A grade less than 4 is a failing grade).

Solution:

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

int main() {
    int grade, sum = 0, count = 0, highest = 0, failCount = 0;
    char choice;

    do {
        cout << "Enter a grade (1 to 9): ";
        cin >> grade;

        if (grade < 1 || grade > 9) {
            cout << "Invalid grade! Please enter a value between 1 and 9." << endl;
            continue;
        }

        sum += grade;
        count++;

        if (grade > highest) {
            highest = grade;
        }

        if (grade < 4) {
            failCount++;
        }

        cout << "Do you want to enter more grades? (Y/N): ";
        cin >> choice;

    } while (choice == 'Y' || choice == 'y');

    if (count > 0) {
        double average = static_cast<double>(sum) / count;
        cout << "Grades average: " << average << endl;
        cout << "Highest grade: " << highest << endl;
        cout << "Number of students failed: " << failCount << endl;
    } else {
        cout << "No grades were entered." << endl;
    }

    return 0;
}
```

This program calculates the average of grades, the highest grade, and the count of failing grades (<4) as the user inputs grades and decides whether to continue.

(b) Use For loop in C++ programming language to generate the shape shown in the screenshot below.

Solution:

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

int main() {
    for (int i = 1; i <= 10; i++) {
        for (int j = 1; j <= i; j++) {
            cout << "*";
        }
        cout << endl;
    }
    return 0;
}
```

2. (a) (i) Use Visual Basic program to construct an interface named "Multiplication table" which consists of command button called "Display Table" and "PictureBox". Set Blue color as background color of the form and adjust the PictureBox control so that it can display multiplication table.

Solution:

1. Open Visual Basic and create a new form.
2. Add the following elements to the form:
   - A `PictureBox` named `PictureBox1`.
   - A `CommandButton` named `cmdDisplayTable` with the caption "Display Table".
3. Set the form's background color to blue using the `BackColor` property.

Visual Basic code for the interface:

```
Private Sub cmdDisplayTable_Click()
```

```
    Dim i As Integer, j As Integer
    Dim result As String

    PictureBox1.Cls ' Clear the PictureBox
    result = "Multiplication Table (1 to 10)" & vbCrLf & vbCrLf

    For i = 1 To 10
        For j = 1 To 10
            result = result & i & " x " & j & " = " & (i * j) & vbTab
        Next j
        result = result & vbCrLf
    Next i

    PictureBox1.Print result
End Sub
```

2. (a) (ii) Generate a program using Visual Basic interface created in (a)(i) above to display multiplication tables for integers from 1 to 4 after clicking a command button "Display Table".

Solution:
Modify the above code to limit the multiplication table to integers from 1 to 4:

```
Private Sub cmdDisplayTable_Click()
    Dim i As Integer, j As Integer
    Dim result As String

    PictureBox1.Cls ' Clear the PictureBox
    result = "Multiplication Table (1 to 4)" & vbCrLf & vbCrLf

    For i = 1 To 4
        For j = 1 To 10
            result = result & i & " x " & j & " = " & (i * j) & vbTab
        Next j
        result = result & vbCrLf
    Next i

    PictureBox1.Print result
End Sub
```

2. (b) (i) Use Visual Basic program to create the Traffic Light interface below:

Interface description:

- Set timer interval to 3000 equivalent to 3 seconds for each light.
- Set Traffic Light colors as Green, Yellow, and Red.
- Set a form so that it cannot be resized by the user.
- Set Active border or &H8000000A& as frame background color.

Solution:

1. Open Visual Basic and create a new form.
2. Add the following components to the form:
   - A `Timer` control named `Timer1`.
   - Three `Shape` controls named `RedLight`, `YellowLight`, and `GreenLight` from the Shape control toolbox.
   - A `Label` control named `lblTitle` with the text "TRAFFIC LIGHT".
3. Adjust the following properties:
   - Timer interval: 3000 (3 seconds).
   - Set the form's `BorderStyle` property to `FixedSingle` to prevent resizing.
   - Set the form's `BackColor` to `&H8000000A&`.

Visual Basic code:

```
Private Sub Form_Load()
    ' Set initial state for the traffic light
    RedLight.FillColor = vbRed
    YellowLight.FillColor = vbWhite
    GreenLight.FillColor = vbWhite

    ' Start the timer
    Timer1.Enabled = True
End Sub

Private Sub Timer1_Timer()
    ' Change the traffic light colors sequentially
    If RedLight.FillColor = vbRed Then
        RedLight.FillColor = vbWhite
        YellowLight.FillColor = vbYellow
    ElseIf YellowLight.FillColor = vbYellow Then
        YellowLight.FillColor = vbWhite
        GreenLight.FillColor = vbGreen
    ElseIf GreenLight.FillColor = vbGreen Then
        GreenLight.FillColor = vbWhite
        RedLight.FillColor = vbRed
    End If
End Sub
```

Steps for the Interface:
- Align the `Shape` controls vertically to represent the traffic lights (Red, Yellow, Green).
- Set the `Shape` type to `Circle`.
- Set their `FillColor` to `vbWhite` initially.
- Label the form appropriately.

(ii) Write Visual Basic codes which will activate the interface created in (b)(i) above to implement like an actual one-way traffic light.

Solution:
The above code achieves the traffic light functionality. When the form loads, the `Timer` control alternates the lights every 3 seconds:
1. Starts with the red light on.
2. Moves to yellow after 3 seconds.
3. Turns green after another 3 seconds, then cycles back to red.

3. Use HTML codes to design webpages for Computer World Solutions Company as shown below. The login link should open a form which asks a user to enter username and password. Validate a form using JavaScript (check whether the entries are blank and username is not a number).

Page description:
- Both table width and height are 100%.
- The width and height of the first row are 100% and 40%, respectively.
- The width and height of the second row are 100% and 60%, respectively.
- Colspan for the first row should be 3.
- Font size and color of the heading is 84 and Maroon, respectively.
- Background color of the heading Compuworld Solutions should be light blue.
- Font color of the statement "Dealers in all IT products" should be blue.
- Background color of the main links, events, and picture should be lavender.
- All links should not open (dead link) except the login link.
- Use your favorite image from the pictures folder available in your computer.

Solution:

```
<!DOCTYPE html>
<html>
<head>
  <title>Compuworld Solutions</title>
  <style>
    body {
      margin: 0;
    }
    table {
      width: 100%;
      height: 100%;
      border-collapse: collapse;
    }
    .header {
      background-color: lightblue;
      text-align: center;
      font-size: 84px;
      color: maroon;
    }
    .subheader {
      color: blue;
      text-align: center;
    }
    .main {
      background-color: lavender;
    }
    a {
      text-decoration: none;
      color: black;
    }
    a:visited {
      color: black;
    }
  </style>
  <script>
    function validateForm() {
      var username = document.forms["loginForm"]["username"].value;
      var password = document.forms["loginForm"]["password"].value;

      if (username == "" || password == "") {
          alert("Username and password cannot be blank.");
          return false;
      }
      if (!isNaN(username)) {
          alert("Username cannot be a number.");
          return false;
      }
      return true;
    }
  </script>
</head>
<body>
```

```
    <table>
        <tr class="header" style="height: 40%;">
            <td colspan="3">Compuworld Solutions™</td>
        </tr>
        <tr class="subheader" style="height: 10%;">
            <td colspan="3">Dealers in all IT products</td>
        </tr>
        <tr style="height: 50%;">
            <td class="main" style="width: 25%;">
                <ul>
                    <li><a href="#">Home</a></li>
                    <li><a href="#">Products</a></li>
                    <li><a href="#">Services</a></li>
                    <li><a href="#" onclick="alert('Login not functional yet')">Login</a></li>
                    <li><a href="#">Customers Logged</a></li>
                </ul>
            </td>
            <td class="main" style="width: 50%;">
                <h3>Events</h3>
                <p>Instock for Online buyers</p>
            </td>
            <td class="main" style="width: 25%;">
                <img src="your_image.jpg" alt="Company Image" style="width: 100%; height: auto;">
            </td>
        </tr>
    </table>
    <a href="#loginForm" onclick="document.getElementById('loginForm').style.display =
'block'">Login</a>
    <div id="loginForm" style="display: none;">
        <form name="loginForm" onsubmit="return validateForm()">
            <label for="username">Username:</label>
            <input type="text" id="username" name="username"><br>
            <label for="password">Password:</label>
            <input type="password" id="password" name="password"><br>
            <input type="submit" value="Submit">
        </form>
    </div>
</body>
</html>
```

This code satisfies all the specifications by using HTML and JavaScript for validation and styling. The login link opens a form that validates the username and password. The design elements align with the requirements given.