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: 2017**

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) Construct a C++ program which takes a character as an input from the keyboard and converts it into a capital letter if it is a small letter and vice versa. A program should check the validity of the entered character and display a message "The entered character is not an alphabetical letter" if the input is not a letter.

Solution:

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

int main() {
   char input;

   cout << "Enter a character: ";
   cin >> input;

   if (isalpha(input)) {
      if (islower(input)) {
         input = toupper(input);
         cout << "Converted to uppercase: " << input << endl;
      } else if (isupper(input)) {
         input = tolower(input);
         cout << "Converted to lowercase: " << input << endl;
      }
   } else {
      cout << "The entered character is not an alphabetical letter." << endl;
   }

   return 0;
}
```

This program checks if the entered character is alphabetical. If true, it toggles its case (uppercase to lowercase or vice versa)

1. (b) Use "For loop", "While loop", and "Do...while loop" to create a C++ program that prints prime numbers up to 31. (HINT: The outputs should be printed on the same screen).

Solution:

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

// Function to check if a number is prime
bool isPrime(int num) {
```

```
    if (num <= 1) return false;
    for (int i = 2; i <= num / 2; i++) {
        if (num % i == 0) return false;
    }
    return true;
}

int main() {
    // Using For loop
    cout << "Prime numbers using For loop:" << endl;
    for (int i = 2; i <= 31; i++) {
        if (isPrime(i)) {
            cout << i << " ";
        }
    }
    cout << endl;

    // Using While loop
    cout << "Prime numbers using While loop:" << endl;
    int i = 2;
    while (i <= 31) {
        if (isPrime(i)) {
            cout << i << " ";
        }
        i++;
    }
    cout << endl;

    // Using Do...while loop
    cout << "Prime numbers using Do...while loop:" << endl;
    int j = 2;
    do {
        if (isPrime(j)) {
            cout << j << " ";
        }
        j++;
    } while (j <= 31);
    cout << endl;

    return 0;
}
```

2. (a) (i) Use Visual Basic program to design the user interface as given below:
Solution for the User Interface:
1. Open Visual Basic and create a new form.
2. Add the following controls:
  - TextBoxes:
    - `txtLoanAmount` for "Loan amount".
    - `txtInterestRate` for "Interest rate".
    - `txtDuration` for "Duration (in months)".
    - `txtPayment` for displaying the monthly payment.
  - Checkbox: `chkEarlyPayment` labeled "Check if early payment".
  - Command Buttons:
    - `cmdShowPayment` labeled "Show Payment".
    - `cmdClear` labeled "Clear".
    - `cmdExit` labeled "Exit".

(ii) Construct Visual Basic codes which will enable a user to calculate monthly payment after entering the loan amount, interest rate, and duration. A program should display the amount required to be paid on the text box after clicking the command button "Show Payment".

HINT: Use the following formula to calculate monthly payment:
`Monthly Payment = Pmt(InterestRate, Duration, LoanAmount, 0, Due)`
Note.
- Due is assigned to 0 if the checkbox is unchecked and to 1 if checked.

Solution for the Code:

```
Private Sub cmdShowPayment_Click()
    Dim LoanAmount As Double
    Dim InterestRate As Double
    Dim Duration As Integer
    Dim Due As Integer
    Dim MonthlyPayment As Double

    ' Get input values
    LoanAmount = Val(txtLoanAmount.Text)
    InterestRate = Val(txtInterestRate.Text) / 100 / 12 ' Convert annual percentage to monthly decimal
    Duration = Val(txtDuration.Text)

    ' Check if early payment is selected
    If chkEarlyPayment.Value = 1 Then
        Due = 1
    Else
        Due = 0
```

```
    End If
    ' Calculate monthly payment
    On Error Resume Next
    MonthlyPayment = Pmt(InterestRate, Duration, -LoanAmount, 0, Due)
    On Error GoTo 0

    ' Display the payment
    txtPayment.Text = Format(MonthlyPayment, "0.00")
End Sub

Private Sub cmdClear_Click()
    ' Clear all inputs
    txtLoanAmount.Text = ""
    txtInterestRate.Text = ""
    txtDuration.Text = ""
    txtPayment.Text = ""
    chkEarlyPayment.Value = 0
End Sub
Private Sub cmdExit_Click()
    ' Close the application
    Unload Me
End Sub
```

Steps for Execution:

1. The `cmdShowPayment_Click` calculates the monthly payment using the formula provided and displays it in `txtPayment`.
2. The `cmdClear_Click` clears all fields and resets the checkbox.
3. The `cmdExit_Click` closes the form.

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

Interface Requirements:

1. Two ListBoxes:
   - lstUnsorted for "Unsorted List".
   - lstSorted for "Sorted List".
2. Buttons for each ListBox:
   - For lstUnsorted: btnRemoveUnsorted, btnAddUnsorted, btnClearUnsorted.
   - For lstSorted: btnRemoveSorted, btnAddSorted, btnClearSorted.
3. TextBox:
   - txtNewElement to input new elements.
4. Additional buttons:
   - btnGo for transfer between lists.
   - btnBack to transfer back.

(ii) Create Visual Basic codes which will activate each command in the interface created in part (b)(i).

Solution for the Code:

```
Private Sub btnAddUnsorted_Click()
    ' Add an element to the unsorted list
    If txtNewElement.Text <> "" Then
        lstUnsorted.AddItem txtNewElement.Text
        txtNewElement.Text = ""
    Else
        MsgBox "Please enter a valid element.", vbExclamation
    End If
End Sub

Private Sub btnRemoveUnsorted_Click()
    ' Remove the selected element from the unsorted list
    If lstUnsorted.ListIndex >= 0 Then
        lstUnsorted.RemoveItem lstUnsorted.ListIndex
    Else
        MsgBox "Please select an item to remove.", vbExclamation
    End If
End Sub

Private Sub btnClearUnsorted_Click()
    ' Clear all elements in the unsorted list
    lstUnsorted.Clear
End Sub

Private Sub btnAddSorted_Click()
    ' Add an element to the sorted list
    If txtNewElement.Text <> "" Then
        lstSorted.AddItem txtNewElement.Text
        txtNewElement.Text = ""
    Else
        MsgBox "Please enter a valid element.", vbExclamation
    End If
End Sub

Private Sub btnRemoveSorted_Click()
    ' Remove the selected element from the sorted list
    If lstSorted.ListIndex >= 0 Then
        lstSorted.RemoveItem lstSorted.ListIndex
    Else
        MsgBox "Please select an item to remove.", vbExclamation
```

```
    End If
End Sub

Private Sub btnClearSorted_Click()
    ' Clear all elements in the sorted list
    lstSorted.Clear
End Sub

Private Sub btnGo_Click()
    ' Transfer the selected item from unsorted to sorted list
    If lstUnsorted.ListIndex >= 0 Then
        lstSorted.AddItem lstUnsorted.List(lstUnsorted.ListIndex)
        lstUnsorted.RemoveItem lstUnsorted.ListIndex
    Else
        MsgBox "Please select an item to transfer.", vbExclamation
    End If
End Sub

Private Sub btnBack_Click()
    ' Transfer the selected item from sorted to unsorted list
    If lstSorted.ListIndex >= 0 Then
        lstUnsorted.AddItem lstSorted.List(lstSorted.ListIndex)
        lstSorted.RemoveItem lstSorted.ListIndex
    Else
        MsgBox "Please select an item to transfer back.", vbExclamation
    End If
End Sub
```

Implementation:

1. Add Element:
   - The Add button adds a new item to the respective ListBox (lstUnsorted or lstSorted) using txtNewElement.
2. Remove Element:
   - The Remove button deletes the selected item from the corresponding ListBox.
3. Clear List:
   - The Clear button removes all items from the ListBox.
4. Transfer:
   - The Go button moves a selected item from lstUnsorted to lstSorted.
   - The Back button moves an item from lstSorted to lstUnsorted.

3. (a) Use basic HTML tags and JavaScript codes to design a website which consists of two webpages namely "Home.html" and "Metric-converter.html". The screenshots of the webpages are as follows:

Page Descriptions:
- The main table should have width=800, height=600, cell padding=2, border=1, and aligned at center.
- The width of Menu buttons and main content column should be 150 and 650 respectively.
- The input box in metric-converter.html is set to read-only (the user can only read from it).

Solution:
Home.html

```
<!DOCTYPE html>
<html>
<head>
   <title>Home Page</title>
   <style>
      body {
         font-family: Arial, sans-serif;
      }
      table {
         width: 800px;
         height: 600px;
         border: 1px solid black;
         border-collapse: collapse;
         margin: 0 auto;
      }
      td {
         padding: 2px;
      }
      .menu {
         width: 150px;
         vertical-align: top;
      }
      .content {
         width: 650px;
      }
      button {
         width: 100%;
         margin: 5px 0;
      }
   </style>
   <script>
      function navigateTo(page) {
         window.location.href = page;
```

```
    }

    function closeWindow() {
      window.close();
    }

    function goBack() {
      window.history.back();
    }

    function goForward() {
      window.history.forward();
    }
  </script>
</head>
<body>
  <table>
    <tr>
      <td colspan="2"><h1 align="center">JAVASCRIPT DRIVEN WEBSITE</h1></td>
    </tr>
    <tr>
      <td class="menu">
        <button onclick="navigateTo('Home.html')">Home</button>
        <button onclick="navigateTo('Metric-converter.html')">Metric Converter</button>
        <button onclick="closeWindow()">Close Window</button>
      </td>
      <td class="content">
        <h2>Homepage</h2>
      </td>
    </tr>
    <tr>
      <td colspan="2" align="center">
        <button onclick="goBack()"><<< Go Back</button>
        <button onclick="goForward()">Go Forward >>></button>
      </td>
    </tr>
  </table>
</body>
</html>
```

Metric-converter.html

```
L<!DOCTYPE html>
```

```
<html>
<head>
  <title>Metric Converter</title>
  <style>
    body {
      font-family: Arial, sans-serif;
    }
    table {
      width: 800px;
      height: 600px;
      border: 1px solid black;
      border-collapse: collapse;
      margin: 0 auto;
    }
    td {
      padding: 2px;
    }
    .menu {
      width: 150px;
      vertical-align: top;
    }
    .content {
      width: 650px;
    }
    select, input {
      width: 90%;
    }
    input[readonly] {
      background-color: lightgray;
    }
  </style>
  <script>
    function navigateTo(page) {
      window.location.href = page;
    }

    function closeWindow() {
      window.close();
    }

    function goBack() {
      window.history.back();
    }
```

```
      function goForward() {
        window.history.forward();
      }
    </script>
</head>
<body>
    <table>
      <tr>
        <td colspan="2"><h1 align="center">JAVASCRIPT DRIVEN WEBSITE</h1></td>
      </tr>
      <tr>
        <td class="menu">
          <button onclick="navigateTo('Home.html')">Home</button>
          <button onclick="navigateTo('Metric-converter.html')">Metric Converter</button>
          <button onclick="closeWindow()">Close Window</button>
        </td>
        <td class="content">
          <h2>Provide Your Choice for Conversion</h2>
          <p>Choose the type of conversion</p>
          <select>
            <option>Select</option>
            <option>Length</option>
            <option>Mass</option>
            <option>Pressure</option>
          </select>
          <h3>Output</h3>
          <input type="text" readonly>
          <button>Refresh Converter</button>
        </td>
      </tr>
      <tr>
        <td colspan="2" align="center">
          <button onclick="goBack()"><<< Go Back</button>
          <button onclick="goForward()">Go Forward >>></button>
        </td>
      </tr>
    </table>
</body>
</html>
```

Explanation:

1. The `Home.html` page includes buttons for navigation, window closing, and browser history control.

2. The `Metric-converter.html` page includes a dropdown for metric conversion options and a read-only output box.
3. Both pages use JavaScript for navigation, history control, and closing the browser window.

(b) Activate "Home" and "Metric Converter" links in the buttons menu by using JavaScript on click mouse event. The links must open a linked page on the same window.

Solution:

This requirement has already been addressed in the **Home.html** and **Metric-converter.html** pages through the following JavaScript code:

```javascript
function navigateTo(page) {
    window.location.href = page;
}
```

The buttons are linked as:

```
html
<button onclick="navigateTo('Home.html')">Home</button>
<button onclick="navigateTo('Metric-converter.html')">Metric Converter</button>
```

These links navigate between the two pages within the same browser window.

(c) Use JavaScript codes to activate the Close Window button.

Solution:

The "Close Window" button functionality has also been implemented using the following JavaScript code:

```
function closeWindow() {
    window.close();
}
```

The button is defined as:

```
<button onclick="closeWindow()">Close Window</button>
```

(d) By using JavaScript codes activate the "Go Back" and "Go Forward" buttons so that they will enable the user to navigate the website through Browser History.

Solution:

The "Go Back" and "Go Forward" buttons are implemented as follows:

```
function goBack() {
   window.history.back();
}

function goForward() {
   window.history.forward();
}
```

The buttons are defined as:

```
<button onclick="goBack()"><<< Go Back</button>
<button onclick="goForward()">Go Forward >>></button>
```

These buttons use the browser's history API to navigate backward and forward through the user's browsing history.

Summary:
1. Navigation between "Home.html" and "Metric-converter.html" is achieved using the `navigateTo()` function.
2. The "Close Window" button is activated using the `closeWindow()` function.
3. The "Go Back" and "Go Forward" buttons use the browser history API (`window.history.back()` and `window.history.forward()`).