Variables and Constants in Visual Basic

Variables

Variables are storage locations in memory that hold data which can be modified during the execution of a program. In Visual Basic, variables must be declared before they can be used.

Rules for Naming Variables

- **Start with a Letter:** Variable names must begin with a letter (A-Z or a-z). They cannot start with a digit or any special character.
- It must be less than 255 characters
- **No Spaces:** Variable names cannot contain spaces. Use underscores (_) or camelCase to separate words if necessary.
- **Use Only Letters, Numbers, and Underscores:** Variable names can include letters, digits (0-9), and underscores (_). They cannot include special characters like @, #, $, %, etc.
- **Avoid Reserved Keywords:** Variable names cannot be the same as reserved keywords in Visual Basic, such as Dim, If, Else, Next, etc.
- **Case Sensitivity:** Visual Basic is not case-sensitive, so myVariable, MyVariable, and MYVARIABLE are considered the same. However, it is a good practice to use consistent casing for readability.

Declaring Variables

In Visual Basic, you must declare the variables before you can use them. To declare a variable, you assign a name to the variable and state its data type. If you fail to do so, the program will run into an error. Variables are usually declared in the general section of the code windows using the **Dim** keyword.

Declaring Variables:

- Variables are declared using the Dim statement.
- The syntax is Dim VariableName As DataType.

Example:

```vb
Dim age As Integer
Dim name As String
Dim salary As Decimal
Dim isActive As Boolean
```

Assigning Values:

- Variables can be assigned values using the = operator.
- The value assigned must be compatible with the variable's data type.

Example:

```vb
age = 25
name = "John Doe"
salary = 50000.75D
isActive = True
```

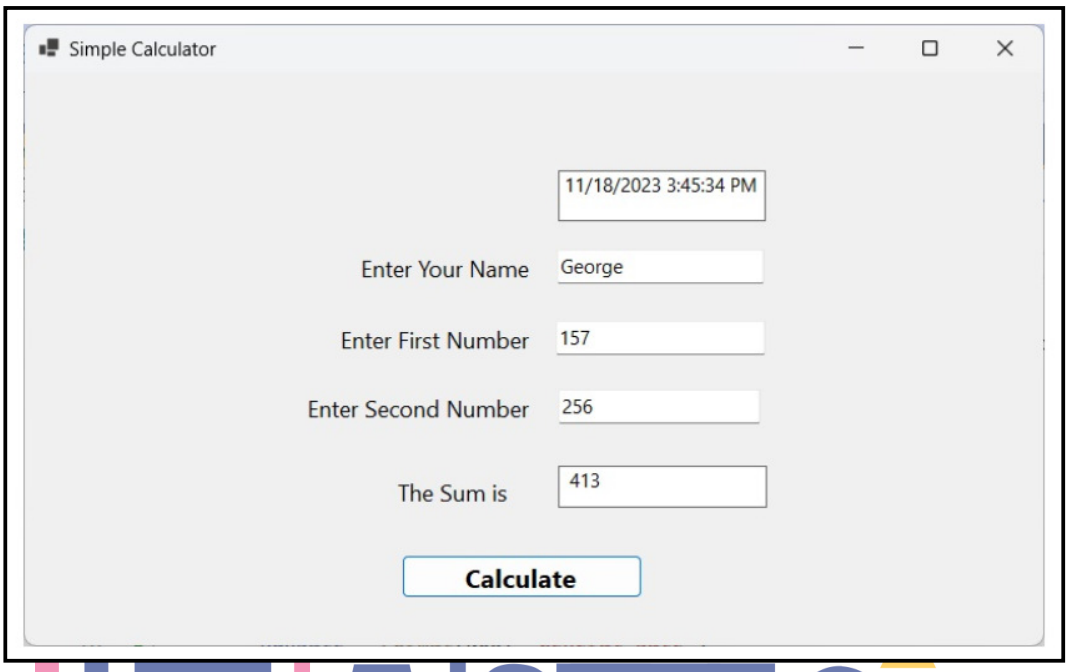
Assignment

Ques 1: Personal Information
Create a form that collects and displays personal information.
1. **Form Design:**
- Add TextBox controls for first name, last name, and age.
- Add a Button labeled "Submit".
- Add a Label to display the full name and age.
2. **Code Requirements:**
- Declare variables to store the first name, last name, and age.
- Assign the values from the TextBox controls to these variables.
- Display the full name and age in the Label when the button is clicked.

Ques 2: Simple Arithmetic
Create a form that performs simple arithmetic operations.
1. **Form Design:**
- Add TextBox controls for two numbers.
- Add Button controls labeled "Add", "Subtract", "Multiply", and "Divide".
- Add a Label to display the result.
2. **Code Requirements:**
- Declare variables to store the two numbers.
- Perform the arithmetic operation based on the button clicked.
- Display the result in the Label.

Ques 3: Shopping Cart Total
Create a form to calculate the total cost of items in a shopping cart.
1. **Form Design:**
- Add TextBox controls for item names and prices (up to 5 items).
- Add a Button labeled "Calculate Total".
- Add a Label to display the total cost.
2. **Code Requirements:**
- Declare variables to store item names and prices.
- Calculate the total cost by summing up the prices.
- Display the total cost in the Label when the button is clicked.

Ques 4: Write the code for this simple calculator.



Visual Basic - Module 06

Constants

Constants are similar to variables, but their values cannot be changed once they are assigned. Constants provide a way to use meaningful names for fixed values that do not change throughout the program.

Declaring Constants:
- Constants are declared using the Const statement.
- The syntax is Const ConstantName As DataType = Value.

Example:

```vb
Const Pi As Double = 3.14159
Const MaxScore As Integer = 100
Const WelcomeMessage As String = "Welcome to Visual Basic!"
```

Using Constants:
- Constants can be used in expressions just like variables.
- Since their values cannot be changed, they provide a safe way to use fixed values.

Example:

```vb
Dim radius As Double = 5.0
Dim area As Double
area = Pi * radius * radius

Dim finalScore As Integer
finalScore = MaxScore - 10
```

Key Points
- **Variables:**
  - Declared using Dim.
  - Can be changed during program execution.
  - Must be assigned a value that matches their data type.
- **Constants:**
  - Declared using Const.
  - Value cannot be changed once assigned.
  - Used for fixed values that do not change.

Visual Basic - Module 06

Assignment

Ques 1: Calculate the Area of a Circle
Write a Visual Basic program that calculates the area of a circle. Use a constant for the value of Pi (π).
Instructions:
1. Create a Windows Forms application.
2. Add a TextBox control for the radius input.
3. Add a Button labeled "Calculate Area".
4. Add a Label to display the calculated area.
5. Declare a constant for Pi (π) with the value 3.14159.
6. When the button is clicked, read the radius from the TextBox, calculate the area using the formula Area=π×radius2\text{Area} = \pi \times \text{radius}^2Area=π×radius2, and display the result in the Label.
Hint: Use the Double data type for the radius and the area.

Ques 2: Calculate Sales Tax
Write a Visual Basic program that calculates the total cost of an item including sales tax. Use a constant for the sales tax rate.
Instructions:
1. Create a Windows Forms application.
2. Add a TextBox control for the item price input.
3. Add a Button labeled "Calculate Total Cost".
4. Add a Label to display the total cost.
5. Declare a constant for the sales tax rate (e.g., 0.07 for 7%).
6. When the button is clicked, read the item price from the TextBox, calculate the total cost using the formula Total Cost=Item Price × (1+Sales Tax Rate) and display the result in the Label.
Hint: Use the Decimal data type for the item price and total cost.

Ques 3: Convert Fahrenheit to Celsius
Question: Write a Visual Basic program that converts a temperature from Fahrenheit to Celsius. Use a constant for the conversion formula's adjustment factor.
Instructions:
1. Create a Windows Forms application.
2. Add a TextBox control for the Fahrenheit temperature input.
3. Add a Button labeled "Convert to Celsius".
4. Add a Label to display the converted temperature in Celsius.
5. Declare a constant for the conversion factor (5/9) and another constant for the adjustment factor (32).
6. When the button is clicked, read the Fahrenheit temperature from the TextBox, convert it to Celsius using the formula Celsius=(Fahrenheit−32) × 5/9, and display the result in the Label.
Hint: Use the Double data type for the temperatures.