|  |
| --- |
| BMC212253408 |
| Intro to Programming |
| Gatluak Thiep |

|  |
| --- |
| Lecturer: Derron Blount  [Date] |

Contents

[1. Understand simple problem specifications and develop appropriate program designs. 2](#_Toc93092125)

[1.1 Design programs using an appropriate methodology to meet a given specification. 2](#_Toc93092126)

[Problem Definition Statement 2](#_Toc93092127)

[User needs 2](#_Toc93092128)

[Low fidelity prototype 3](#_Toc93092129)

[High Fidelity prototype 4](#_Toc93092130)

[Flow chart 5](#_Toc93092131)

[1.2 Produce data tables of identifiers with clear names and data types. 5](#_Toc93092132)

[1.3 Justify the choice of data types 6](#_Toc93092133)

[1.4 Pseudo code 6](#_Toc93092134)

[2.0 Demonstrate the ability to use basic elements of a programming language 7](#_Toc93092135)

[2.1 Write a program including input and output commands 7](#_Toc93092136)

[Input commands 7](#_Toc93092137)

[3.0 Produce and follow a test plan 7](#_Toc93092138)

[3.1 Devise a test plan with a clear explanation for the choice of test data and justification of choices 7](#_Toc93092139)

[3.2 Produce an associated test log 7](#_Toc93092140)

[3.3 Evaluate the effectiveness of the test plan 8](#_Toc93092141)

[4.0 Evaluate a program 9](#_Toc93092142)

[4.1 Critically evaluate the effectiveness of the program produced. 9](#_Toc93092143)

# Understand simple problem specifications and develop appropriate program designs.

## Design programs using an appropriate methodology to meet a given specification.

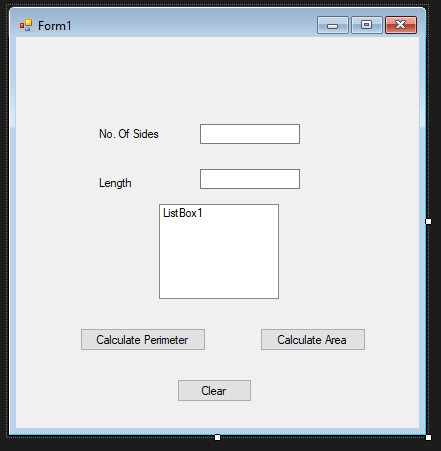
### Problem Definition Statement

The purpose of the program is to find the perimeter and area of a regular polygon using the number of sides and the length of one side to make the calculations. The user needs to enter the number of sides and the length of one side and that should determine the perimeter but to find the area the perimeter needs to be multiplied by the apothem. When the calculate area button is clicked that will trigger a calculation to give the answer that will then be displayed in a list box.

### User needs

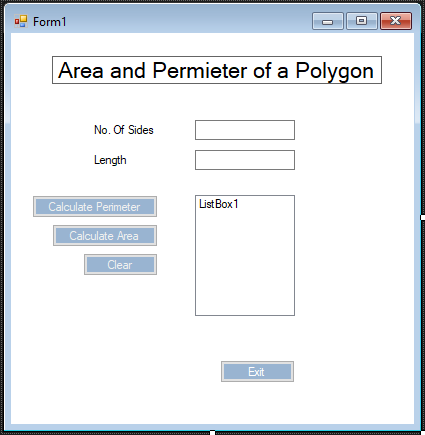
The user needs to have a place to enter the values needed to calculate the answer. Also needs buttons to press in order to trigger the calculation as well as a button to clear the values entered in the text box and a button to exit the program

### Low fidelity prototype



*Figure 1.11 is a screenshot of the low fidelity prototype.*

### High Fidelity prototype



*Figure 1.12 is a screenshot of the high fidelity prototype*

### Flow chart

Diagram

Description automatically generated

*Figure 1.13 shows the flowchart of the program*

## Produce data tables of identifiers with clear names and data types.

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Global or Local |
| noOfSides | integer | local |
| length | double | global |
| perimeter | double | global |
| Apothem | Integer | global |
| Area | Double | Local |
| Answer | Double | Global |

## Justify the choice of data types

noOfSides – integer because it’s a whole number less than 2,000,000 – local because will only be used to when figuring out the perimeter

length – double because it’s a number but it could have decimals – global because it will be used multiple times

perimeter – double because the equation could result in a decimal and it’s a number – global because it will be used multiple times throughout the code.

Apothem – double as it will most likely have a decimal place in it since Pi is used to solve the equation – global because it will be used to solve another problem

Area – double because it will be a number that could return decimal places

### 1.4 Pseudo code

START PROGRAM

ENTER number in noOfSides textbox

ENTER number in length textbox

PRESS calculate perimeter THEN

SHOW perimeter = in listbox

PRESS calculate area THEN

SHOW area = in listbox

END program

# 2.0 Demonstrate the ability to use basic elements of a programming language

## 2.1 Write a program including input and output commands

### Input commands

to calculate perimeter:

Text

Description automatically generated

*Figure 2.11 shows code to calculate the perimeter.*

To calculate area:



*Figure 2.12 shows how to figure out the apothem.*

**

*Figure 2.13 shows the area being calculated using the perimeter and apothem.*

### Output commands



*Figure 2.14 shows the output command for perimeter into the listbox.*

**

*Figure 2.15 shows the output for a error message.*

**

*Figure 2.16 shows the output for the area of the polygon.*

# 3.0 Produce and follow a test plan

## 3.1 Devise a test plan with a clear explanation for the choice of test data and justification of choices

I am going to be testing my simple program to see if when numbers are entered, they return the results that I want it to. Since I am using mathematical equations, my focus is going to be to make sure that the program returns the correct answer. I used the waterfall model to build this program as it was a simple program that did not need many versions in order to build.

## 3.2 Produce an associated test log

|  |  |  |  |
| --- | --- | --- | --- |
| Feature | Test cases | Expected result | Actual result |
| Multiplying the no. of sides by the length to get the perimeter | 1st – 4 \* 4  2nd – 6 \* 7 | 16  42 | 16  42 |
| Multiplying the perimeter by apothem to get the area | 1st- perimeter \* apothem | Correct answer | Wrong answer (too many decimals, had to change the code |
| Listbox to show names and values when buttons pressed | Pressed “calculate perimeter” and “calculate area” after values entered | For values with text to be entered into the listbox | Values shown but with no indication what they were so had to include “perimeter = ()” and “area= = ()” in the code. |
| Clearing of all values entered on the form when “clear” button pressed | Press the “clear” button after calculating perimeter and area | All items to be cleared | All items in the text boxes and list box were cleared |
| Error checking so that only numbers can be entered, if numbers was not entered then a message box would come up | 1st – entered a letter in the no. of sides textbox  2nd – entered a letter in the length textbox | Error message to be displayed in a messagebox.  Error message to be displayed in the messagebox. | Error message displayed  Error message displayed |
| Error checking to see what happens when nothing is entered into the program | Pressed calculate perimeter with nothing into the textboxes | Nothing to happen | Program crashed |
| Performance check for program load | Loaded fully complete program | Take a few seconds | Took 5 seconds to load |
| Error message checking for more than 2 sides | 1st – enter 2 into the number of sides text box | Message box shown | Error message shown and could not calculate area because no polygon has less than 3 sides |
| Performance check for time needed to complete calculation | Entered sides and length and pressed ”calculate perimeter” and “calculate area” | Expected to taker less than 15 seconds total as you only need to enter 2 values and press two buttons | Took about 10 seconds with pauses to read the results. |

## 3.3 Evaluate the effectiveness of the test plan

I chose to test the mathematical equations in the program first because that was what the program is intended to do, find the area and perimeter of a regular polygon, so I made sure that worked first and then proceeded to add different features form there. First, I needed to create an equation that calculated the perimeter so that I could find the area. Also, I had to input code to ensure that the apothem was calculated when I pressed the “calculate area” button as it was necessary to have the formula in order to calculate the area.

Next, I created and tested a listbox so that the answers could be shown on it which was a requirement in the design specification.

I created a clear button so that the user could start from scratch and then I put in some code to show an error message whenever someone entered a letter instead of a number.

I believe the test plan was effective because it helped me try different lines of code in order to make the program work as I had intended. There were a lot more errors than shown on the test log however, I was not logging as I was testing which is something I will do moving forward in order to be more thorough in my assignments and so that I can recall the errors I made in order to not do them again and learn from it.

# 4.0 Evaluate a program

## 4.1 Critically evaluate the effectiveness of the program produced.

The program I produced was very straightforward as it was a program that just needed to return two values on the listbox. It is very user friendly as it has simple buttons and clear labels to show what to do in the program. When calculating the perimeter and area all the user needs to do is press the button of the intended answer they want. The program only had a few triggers however, it was challenging to get the apothem calculation correct because I needed to figure out how to make the calculation while using code. Calculating the perimeter was simple as it was just multiplying two values.

In the future I will do testing after every iteration because with this program I was constantly testing different methods however I did not think to record the data until I almost had a complete program. Also the design of the program could have been a bit better by including different colors in the program.

# Code

Public Class Form1

Private Sub btnPer\_Click(sender As Object, e As EventArgs) Handles btnPer.Click

Dim length, noOfSides, perimeter As String

'used as string and then converted to its decimal equivalent number

If noOfSides <= 2 Then

MessageBox.Show("Please enter more than 2 sides")

End If

Try

length = Decimal.Parse(TextBoxLength.Text, Globalization.NumberStyles.Number)

noOfSides = Decimal.Parse(txtBoxSides.Text, Globalization.NumberStyles.Number)

'tryparse parses the string into decimal

'perimeter + length x number of sides

perimeter = Val(length) \* Val(noOfSides)

'listbox used so that multiple elements can show in the listbox

ListBox1.Items.Add("Perimeter = " & perimeter)

Catch

MessageBox.Show("Please enter a number")

End Try

End Sub

Private Sub btnArea\_Click(sender As Object, e As EventArgs) Handles btnArea.Click

Dim length, noOfSides, perimeter, area As Double

If noOfSides <= 2 Then

MessageBox.Show("A polygon must have more than 2 sides")

End If

Try

length = Decimal.Parse(TextBoxLength.Text, Globalization.NumberStyles.Number)

noOfSides = Decimal.Parse(txtBoxSides.Text, Globalization.NumberStyles.Number)

Dim Apothem As Double = TextBoxLength.Text / (2 \* (Math.Tan(Math.PI / txtBoxSides.Text)))

'area = apothem x perimeter / 2

perimeter = (length) \* (noOfSides)

area = (perimeter \* Apothem) / 2

ListBox1.Items.Add("Area = " & Math.Round(area, 2))

'math.round 2 will round the equation to two decimal places

Catch

MessageBox.Show("Please enter a number")

End Try

End Sub

Private Sub btnClear\_Click(sender As Object, e As EventArgs) Handles btnClear.Click

TextBoxLength.Clear()

txtBoxSides.Clear()

ListBox1.Items.Clear()

' to clear all of the items when the clear button is pressed

End Sub

Private Sub btnExit\_Click(sender As Object, e As EventArgs) Handles btnExit.Click

Me.Close()

'to close the program

End Sub

End Class