Programming Fundamentals

Problem Set 1

Instructor:
Dr. Samyan Qayyum Wahla

Learning Objectives:

- Analysis of problem statement
- Installation of RAPTOR(Flow charting tool)
- Describe algorithm in the form of pseudo code and flow chart

CLO:

- CLO1

Registration Number:

_____

Name:

_____

Flow charts:

One of great things about flowcharts is that they explicitly show the logic and flow of control of the algorithm that they express. However, the narrative and verbal descriptions of the problems that they are designed to solve can be much more ambiguous. Problem descriptions, which represent the requirements for your program, may have omissions and inconsistencies and may hide some important aspects of the problem. Thus we must learn to read the requirements of a problem statement with a critical eye and turn the requirements into a more explicit format.

This brief reading will give you some simple methods and tools to help you identify and analyze the requirements within a problem statement. Here's a quick summary:

- Make an itemized list of the requirements as you read the problem statement.
- Explicitly list the inputs, outputs and major processing steps included in the problem description.
- When a condition is specified, make sure you know how to handle every possibility for the variable being tested
- When conditions involve multiple variables, make a table with the rows representing conditions for one variable and the columns representing the conditions for the other variable(s). Then fill in the table to state what action should be performed for each combination of conditions.

An Example – How to create an algorithm

This section will walk you through the process of creating an algorithm for a simple problem. Hopefully you will use the principles discussed here when it comes time for you to develop your own algorithms for different tasks.

Sample Problem: Two people work for the same company and have a total combined salary of W dollars, but one person earns Q dollars more than the other person. How much did they each earn?

Principle 1: Solve a *specific instance* of the problem *by hand* **using pencil and paper.**

A specific instance of a problem is when all the "variables" are given a specific value. Suppose we let W be $57.00 and let Q be $5.00. Now the problem reads:

Specific Problem Instance: Two people work for the same company and have a total combined salary of $57.00, but one person earns $5.00 dollars more than the other person. How much did they each earn?

There are multiple ways to think about this problem. Two possible solutions are:

Arithmetic Reasoning:	**Algebraic reasoning:**
Of the $57, Person1 gets $5 more then Person2. Therefore, both equally share $57-$5 = $52. $52 / 2 is $26, so each gets at least $26. Person1 gets $31 ($26 + $5) Person2 gets $26	Let X = Person1's pay Let Y = Person2's pay. Therefore, we know from the problem that: X = Y + 5 and X + Y = 57. Solve these 2 equations using substitution: (Y + 5) + Y = 57 Y = 26 Therefore X = Y + 5 = 31

Note:
If you cannot solve this problem manually, it is impossible for you to write an algorithm to solve the problem!

Principle 2: Generalize your solution by replacing your specific instance values with "variables".

The two previously shown solutions now become:

Arithmetic Reasoning:	**Algebraic reasoning:**
Of the total salary W, Person1 gets Q dollars more then Person2. Therefore they both equally share (W - Q). Each gets at least (W – Q) / 2. Person1 gets (W – Q) / 2 + Q Person2 gets (W – Q) / 2	Let X = Person1's pay Let Y = Person2's pay. Therefore, we know from the problem that: X = Y + Q and X + Y = W. Solve these 2 equations using substitution: (Y + Q) + Y = W Y = (W – Q)/2 ThereforeX = (W – Q)/2 + Q

Principle 3: Manually execute your algorithm on several test cases to verify that it produces correct answers. This is called *desk checking* or *walking through the algorithm*.

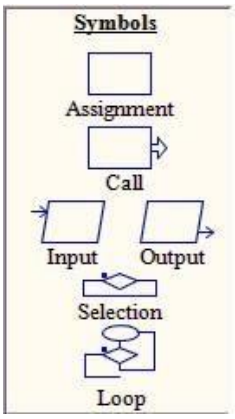
.

Example 1: W = 100, Q = 10
Person1 = (100 – 10)/2 + 10 = 55
Person2 = (100 – 10)/2 = 45
(This is correct because 55 + 45 = 100 and Person1 earns 10 more dollars than Person2)

Example 2: W = 31, Q = 2
Person1 = (31 – 2)/2 + 2 = 16.5
Person2 = (31 – 2)/2 = 14.5
(This is correct because 16.5 + 14.5 = 31 and Person1 earns 2 more dollars than Person2)

Principle 4: After a general solution to the task is known, write correct programming statements to implement your solution on the computer.

RAPTOR



RAPTOR is a flowchart-based programming environment, designed specifically to help students visualize their algorithms and avoid syntactic baggage. RAPTOR programs are created visually and executed visually by tracing the execution through the flowchart. Required syntax is kept to a minimum. Students prefer using flowcharts to express their algorithms, and are more successful creating algorithms using RAPTOR than using a traditional language or writing flowcharts without RAPTOR.
The six symbols used in Raptor are displayed in the Symbol Window in the upper left corner of the main window:

Assignment: The assignment symbol is used to change the value of a variable. The right hand side of the assignment is evaluated, and the resulting value is placed in the variable on the left hand side. For example, consider the case where the value of x is currently 5, and the assignment "x <- x + 1" is executed. First "x+1" is evaluated, yielding the result 6. Then the value of x is changed to be 6. Note that assignment is very different from mathematical equality. The statement should be read "Set x to x+1" instead of "x equals x+1". The assignment symbol can also be used to assign a string expression to a string variable

Call: The call symbol is used to invoke procedures such as graphics routines and other instructor-provided procedures. The call symbol is also used to run subcharts included in a Raptor program. This will not be used in the tasks in this lab manual.

Input: The input symbol is used to ask the user for a number or string while the flowchart is executing. When an input symbol is executed, the user will be prompted with a dialog to enter a value that can be interpreted as either a number or string, depending on what the user types (see String vs. Numeric Input).

Output: The output symbol is used to either write a number or text to the Master Console window.

Selection: The selection structure is used for decision making. The programmer enters in the diamond an expression that evaluates to Yes (True) or No (False). Such expressions are formally referred to as Boolean expressions. Based on the result of the expression in the diamond, control of the program will branch either left (Yes, or True) or right (No, or False). For more information, see Boolean Expressions.

Loop Control: The loop structure is used to repeat a sequence of symbols until a certain condition is met. When execution reaches the bottom of the loop, it starts over again at the top. The loop is exited when the diamond symbol is executed and the Boolean expression in the diamond evaluates to Yes (True).

Installation of RAPTOR:

RAPTOR can be downloaded from the following link https://raptor.martincarlisle.com/raptor2019.msi and can be installed using simple procedure.

Using the toolbar

The toolbar implements many of same functions found in the main menu or in pop-up menus. Hovering the mouse over a single button will display the tooltip for that button.

 The New button creates a new flowchart.

 The Open button opens a previously saved flowchart from disk.

 The Save button saves the current flowchart to disk.

 The Cut, Copy, and Paste buttons all function as expected on the currently selected symbols (those currently colored red).

 The Print button prints the current flowchart.

 The Undo button reverts the flowchart back before the last change was made.

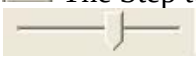
 The Redo button cancels the previous undo operation.

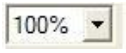
 The Execute to Completion button runs the entire program until it terminates.

 The Pause button temporarily halts execution of a program until execution is resumed by the user.

 The Stop and Reset to Beginning button halts program execution and clears the value of all variables.

 The Step to Next Shape button executes one shape in a flowchart.

 The slider bar on the toolbar adjusts the speed of execution of a flowchart. Move the bar to the left to slow execution. Move the bar to the right to speed up execution. When the slider is positioned to the far right, Raptor will no longer highlight in green the symbol currently being executed. This makes the execution run much faster. To run even faster, use the view menu to turn off the display of variables in the watch window. For fastest execution, use a compiled version of your program.



The drop down box on the right of the toolbar adjusts the scaling factor for the Workspace window.

Building a flowchart

To build a flowchart, left click on a symbol in the Symbol Window. Move the mouse to the place in the flowchart where the symbol belongs. You may need to move the cursor slightly to find an insertion point.

When the cursor is at an insertion point, the cursor will change to  . When the left mouse button is clicked at an insertion point, the selected symbol is added to the existing flowchart at the specified location.
If symbols are incorrectly placed, select Undo, Delete, or Cut from the Edit or Right Mouse Button menu. Symbols may also be copied and pasted in another area of a flowchart.
Once a symbol has been correctly placed, double click the symbol to edit its contents.

Executing a flowchart

A flowchart may be executed using the Run Menu/Execute option or the toolbar. The symbol being executed is highlighted in green. Any variables changed by the execution of a symbol are highlighted in red in the Watch window.

 Step executes through one shape in a flowchart. Pressing F10 will also execute the current shape.

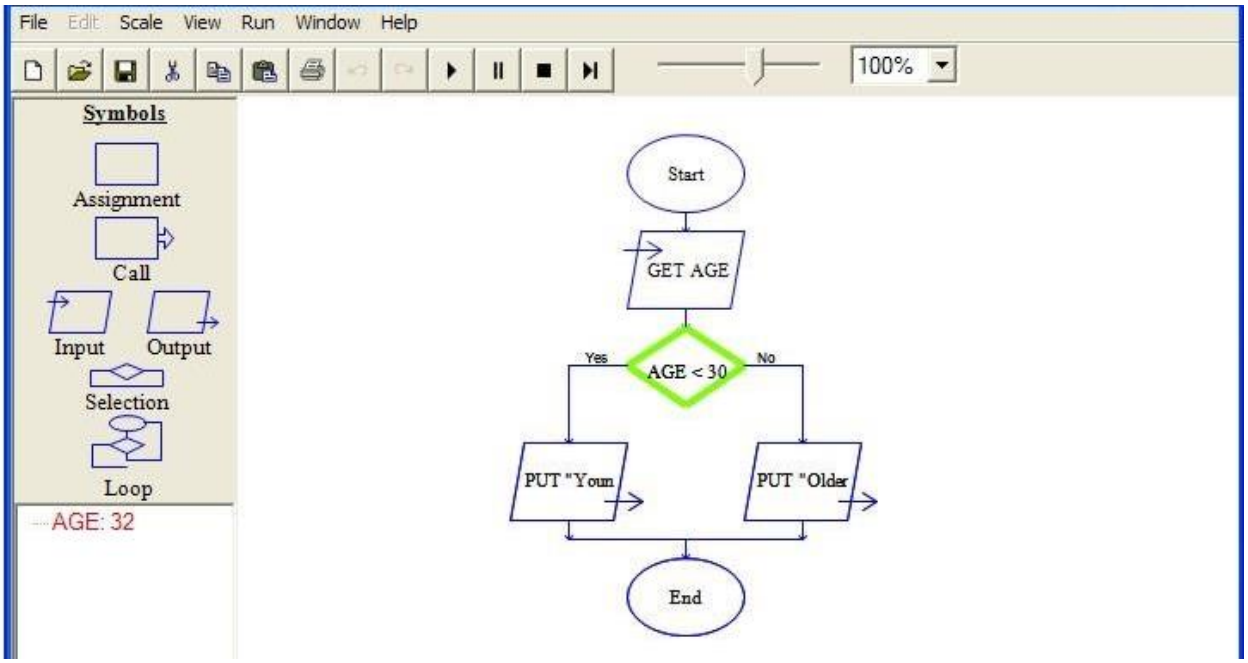
 Execute to Completion runs the entire program until it terminates.

 Reset halts program execution and clears the value of all variables in preparation for a subsequent execution.

Reset/Execute halts program execution, clears the value of all variables, and restarts execution from the beginning. Pressing F5 will also perform this function.

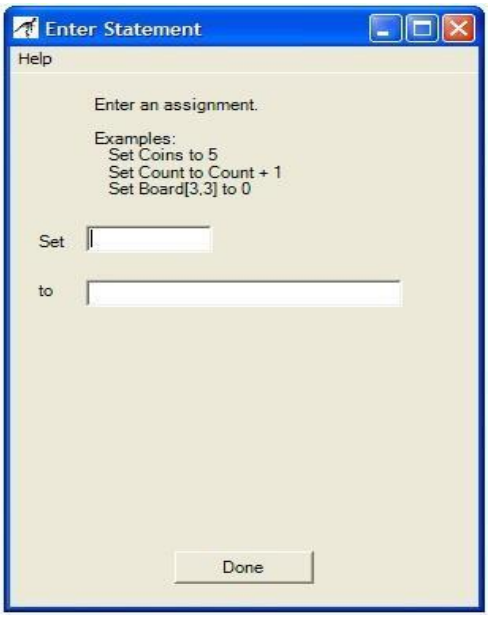
 Pause temporarily halts the execution of a program until execution is resumed by the user.

 Speed Slider sets the speed of program execution. When the slider is entirely to the right (maximum speed), the Watch Window is not updated and flowchart symbols are not highlighted during execution.



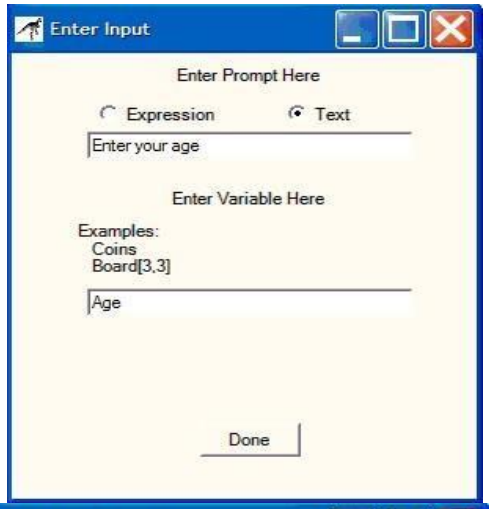
Editing Symbols: Double click a symbol placed in a flowchart to edit its contents.

Assignment: The following dialog appears when editing an assignment symbol.



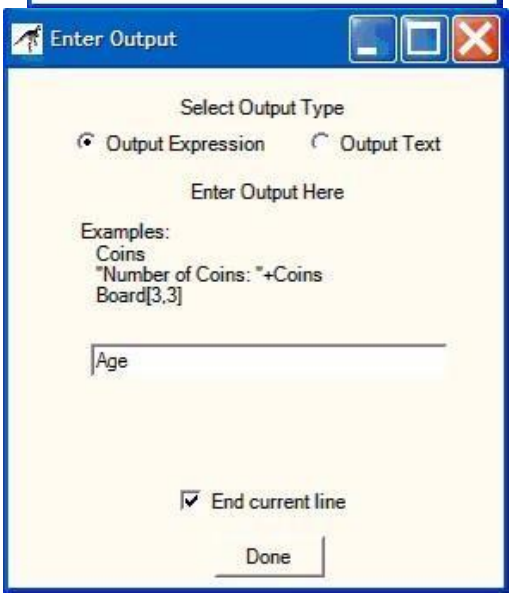
In the box following "Set", enter a variable. In the box following "to", enter an expression whose value will be stored in the given variable. If the variable is a string variable, the expression on the right must be a string expression.

Input Symbol
The following dialog appears when editing an input symbol:



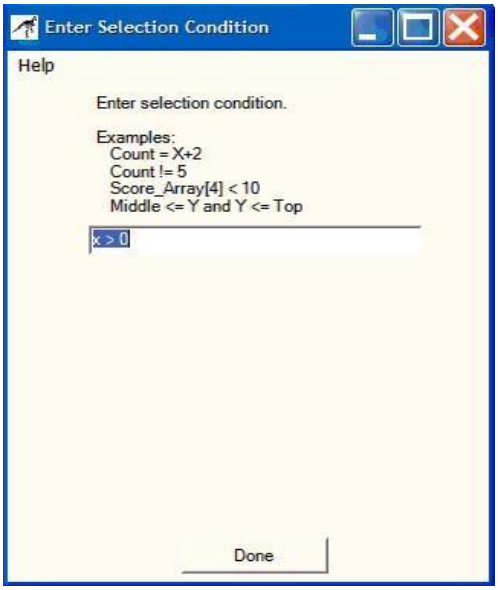
Enter a prompt to the user in the first textbox. The prompt will appear in the pop-up window in which the user will enter a value for the variable whose name is specified in the second textbox. You may select the Expression radio button if you want the prompt to be a string expression that contains a string variable or concatenation of several string expressions. Depending on what the user enters during program execution, the variable (in the above example, Age) may be treated as a number or string.

Output Symbol
The Output Symbol specifies what will be output to the Master Console (or, in some cases, to a file; see Outputting to a File). First select whether an Expression or Text is to be output. An expression can be a number, a string literal in double quotes, a variable, a math expression, or a string expression. Select Text to output a text message containing no variables or math; no quotes are required. If the "End current line" checkbox at the bottom of the dialog is checked, the MasterConsole output after this one will start on the next line.



Selection Symbol

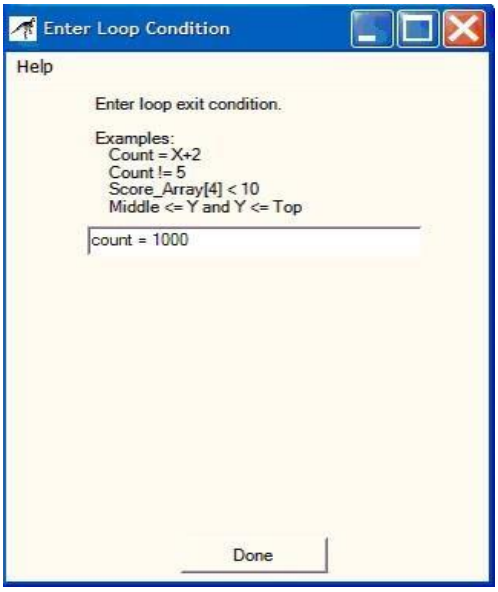
The following dialog appears when editing the decision diamond for a selection structure:



Enter in this textbox a Boolean expression that will be evaluated for this symbol. If the Boolean expression is true, the left branch of the selection will be taken. Otherwise, the right branch will be taken.

Loop Symbol

The following dialog appears when editing the decision diamond for a loop structure:

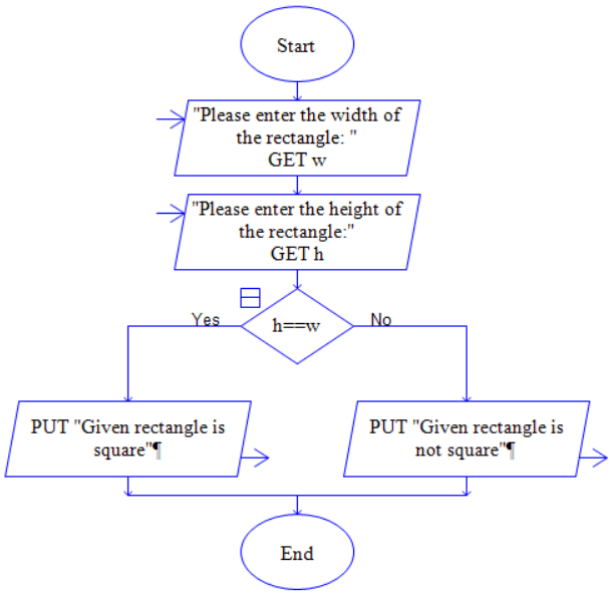


Enter in the textbox a Boolean expression that will be evaluated for this symbol. If the Boolean expression is true, the loop is exited. Otherwise, flow continues on the "No" branch below the diamond and eventually to the top of the loop.

Lets Dive In:

Am I Square: Take values of height and width of a rectangle from user and print on the screen whether it is square or not.

- Draw following flowchart in raptor.
- Click on the execute button in the toolbar
- Check the execution of flowchart on different type of positive and negative integers.
- Is it valid to have the height and width as negative number? Write answer here.



Greatest of the Two: Take two int values from user and print greatest among them.

- Complete the following information before you proceed

Input	x,y
Output	
Format of the output	
Computations	
Test cases	X | y | Expected output | | | | | | | | |

- Write pseudo code below and draw flow chart and then execute it in RAPTOR.

Payable to Shop: A Book Shop has announced 10% of discount on the total purchased amount if it is greater than 1000. Let suppose each book cost 100. Ask the number of purchased book, calculate and print total amount payable to the shop.

- Complete the following information before you proceed

Input	n(number of purchased books)
Output	
Format of the output	
Computations	
Test cases	N | Expected output | | | | | |

- Write pseudo code below and draw flow chart and then execute it in RAPTOR.

Your Turn:

Employee Bonus: A company decided to give bonus of 5% to employee if his/her year of service is more than 5 years. Ask user for their salary and year of service and print the net bonus amount.

Student Attendance: A student will not be allowed to sit in exam if his/her attendance is less than 75%. Take following input from user: Number of classes held and Number of classes attended. Output following: percentage of class attended and Is student is allowed to sit in exam or not.

Greatest of the three: Take three number as input and print the largest number as output with message.

Hint: Use the strategy to solve this problem given in point 4 on page 1.

Problems:

1. Draw a flow chart to check whether a number N is divisible by 3? Take input n from user.
2. Draw flow chart. Take radius r as input from user and calculate its Area according to following formula.

$$A = \pi r^2$$

3. Draw a flow chart to solve the quadratic equation of the form $ax^2 + bx + c$. Take *a,b,c* as input from user. Determine whether the roots are real or imaginary. In case of Imaginary root, just output roots are imaginary. In case of real roots, output the value of roots.
4. Draw a flow chart to display the numbers upto 100 which are divisible by 4. No input should be taken from user.
5. Modify the above Question, Take input from user for N. Display number up to N instead of 100.
6. Draw a flow chart to calculate the sum of numbers between two limits, lower and upper limits. Take lower and upper limit from user as input.
7. Draw a flow chart to calculate the sum of positive numbers entered by user. Take multiple inputs from user unless user enters -1. In case user enters any other negative number, do not add it just ignore it.
8. Draw a flow chart to calculate the factorial of number N entered by user.
9. Draw a flow chart to calculate the Fibonacci of Number N entered by user.
10. Draw a flow chart to display the Fibonacci series up to N. e.g. If N=5, then Fibonacci series should be 0,1,1,2,3.
11. Draw a flowchart to calculate sum of odd digits of a number N.
12. Take input n from user. Draw a flowchart to display the table of number n upto 10.
13. Draw a flow chart to display the series of squares of numbers upto N. e.g. if N = 4 then output should be 1 2 4 9 16
14. Take multiple numbers from user as input unless user enters -1. Other negative numbers are not allowed as input. Draw a flow chart to find the maximum number.
15. Take multiple numbers from user as input unless user enters -1. Draw a flow chart to find the second highest number.
16. Take a number B form user as input. Draw a flow chart to find the reverse of B and output B*2. E.g. if B = 142 then output should be 241*2 = 482
17. Take a number A as input from user, assume it as binary number. Convert binary number to decimal. If A= 101 then output should be 5, if A= 1111 then output should be 15.
18. Draw a flow chart for GCD of two numbers a,b.
19. Draw a flow chart for LCM of two numbers a,b.
20. Draw a flowchart to find out whether a number N is prime or not.

What to submit:

You are required to save the files of given 20 problems in a folder and submit on eduko in Lab Manual 2. All other tasks will be evaluated in Lab.