GERMANNA
ACADEMIC CENTER
FOR EXCELLENCE

Pseudocode and Flowcharts

An often overlooked aspect of software design is the planning stage that must occur before any code is written. The pseudocode and flowchart are integral to this part of the process and for minimizing the number of missteps that can occur when determining how to approach an assignment. This handout will use an example prompt to explain and demonstrate the process of writing pseudocode and creating a flowchart.

You can navigate to specific sections of this handout by clicking the links below.

Pseudocode

Pseudocode informally describes the step-by-step process that will be implemented to solve a problem or accomplish a task. It is useful for planning the logic behind each step of a program and forces the programmer to think critically about what must be accomplished. Therefore, it should always be written before the program code. The pseudocode's informal description allows non-programmers to understand how the program should operate.

When beginning to develop pseudocode, it is important to think about the steps that are needed to solve the given problem. For example, a programmer may be tasked with solving the following prompt:

> Create a program that prompts a user to enter the length and width of a rectangle and then calculates the area of the rectangle based on the user's input. If the area is greater than 100 square units, a message should be displayed that states, "That is a large rectangle." Otherwise, the message should state, "That is a small rectangle."

In programming, the order in which problems are usually solved is to first define the variables, then perform necessary calculations, and finally, display outputs. Problems that are more complex will involve more steps but will likely follow the same general order. In the example prompt, the variables, which are the length and width, are defined by the user's input.

Therefore, the first two steps of the pseudocode must involve prompting the user to enter the length and width of the rectangle. The beginning of the pseudocode will be the following:

```
Define the length by asking for user input
Define the width by asking for user input
```

After the length and width are defined, the area of the rectangle needs to be calculated. The formula for the calculation will be the length multiplied by the width (length * width); it is a good practice to include the formula for steps that involve calculations in the pseudocode. The next step of the pseudocode is shown below.

```
Calculate the area (length * width)
```

After the area of the rectangle has been calculated, the program can then determine the correct output. The output will depend upon the area, so a decision must be made. An “If-Else” statement will be used to determine what the output should be. An “If-Else” statement is a conditional statement that will provide different outcomes based on whether the given condition is true or false. For example, if the statement “The area is greater than 100 square units” is true, the outcome of the condition will be different than if the same condition is false. In the context of the example prompt, if the statement is true, then a message should be displayed that states, “That is a large rectangle.” If the statement is false, the message should instead state, “That is a small rectangle.” The final steps of the pseudocode demonstrate the If-Else statement:

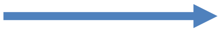
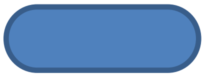
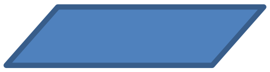
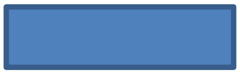
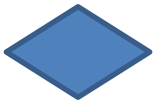
```
If the area is greater than 100 square units
     Display “That is a large rectangle.”
Else
     Display “That is a small rectangle.”
```

Flowchart

The flowchart is used to represent program flow and, much like pseudocode, should be created before any code is written. It utilizes shapes to depict the different processes that occur throughout a program.

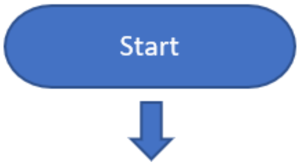
The table shown below features the shapes that are used in a flowchart and the actions that they represent.

Symbols Used In Flowchart

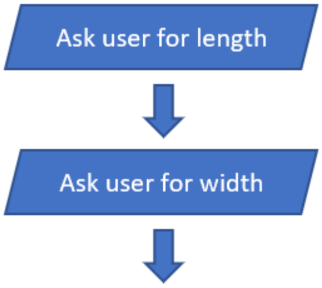
Symbol	Purpose	Description
	Flow line	Indicates the flow of logic by connecting symbols.
	Terminal (Stop/Start)	Represents the start and the end of a flowchart.
	Input/Output	Used for input and output operation.
	Processing	Used for arithmetic operations and data-manipulations.
	Decision	Used for decision making between two or more alternatives.

Flowchart in programming (n.d.). Programiz. https://www.programiz.com/article/flowchart-programming

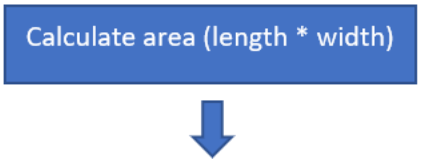
The flowchart for this example will follow the same order as the pseudocode. Every flowchart must begin and end with the terminal shape to signify the beginning and end of the program's operation. Additionally, a flow line should be added between shapes to demonstrate how the program will flow.



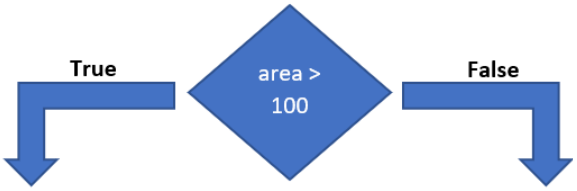
The first two steps of the pseudocode, "Define the length by asking for user input" and "Define the width by asking for user input," are both prompting the user to enter data and are therefore considered inputs. As shown in the image that depicts the flowchart symbols, the shape that represents input and output is a parallelogram, so two parallelograms will follow the beginning terminal shape.



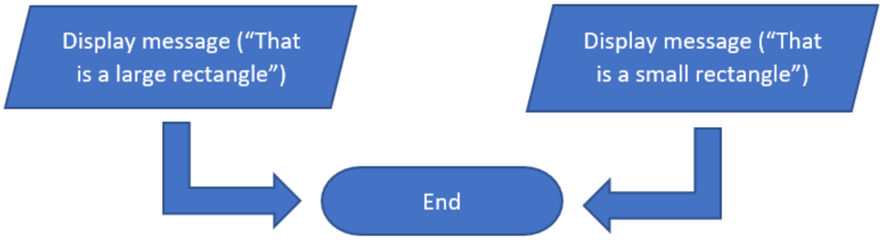
The next step of the program is to calculate the area, which would be considered a process because it is an arithmetic operation. This step will be depicted in a rectangle.

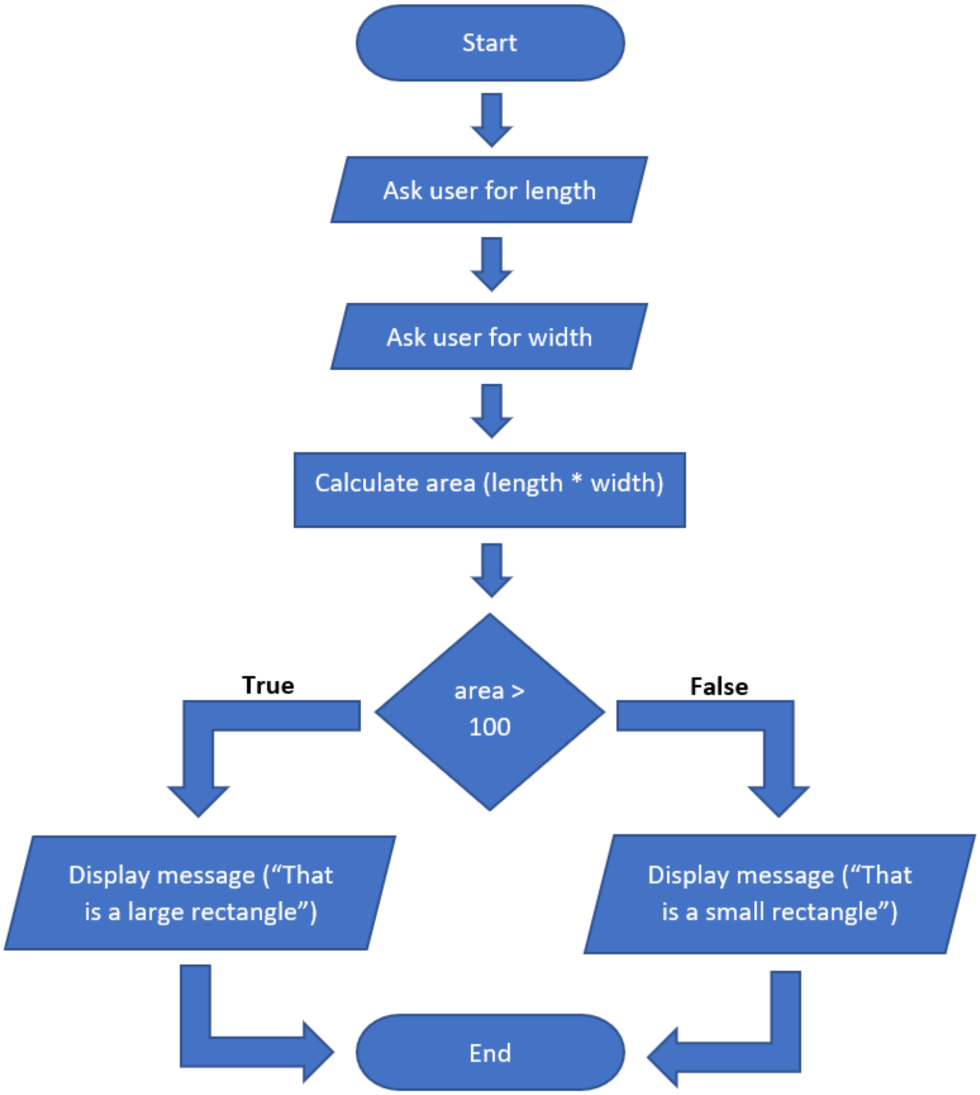


After the area is calculated, a decision must be made regarding whether the condition is true or false. In this example, the condition is if the size of the area is greater than 100. Decisions are depicted by having two pathways emerge from the diamond shape: one path represents the events that occur if the condition is true, while the other shows the results of the condition being false.



As stated in the prompt, a message should be displayed after the decision is made. Regardless of the area of the rectangle, the shape for output will be placed in both the “true” and “false” pathway, and then lead to the end. The finished flowchart for the example prompt is shown on the next page.


Display message (“That is a large rectangle”)
Display message (“That is a small rectangle”)
End


Start
Ask user for length
Ask user for width
Calculate area (length * width)
area > 100
True
False
Display message ("That is a large rectangle")
Display message ("That is a small rectangle")
End

The Academic Center for Excellence (ACE) offers free on-campus or online tutoring appointments for software design. For further assistance with programming concepts, please call ACE at (540) 891-3017 or email us at ACE@germanna.edu.