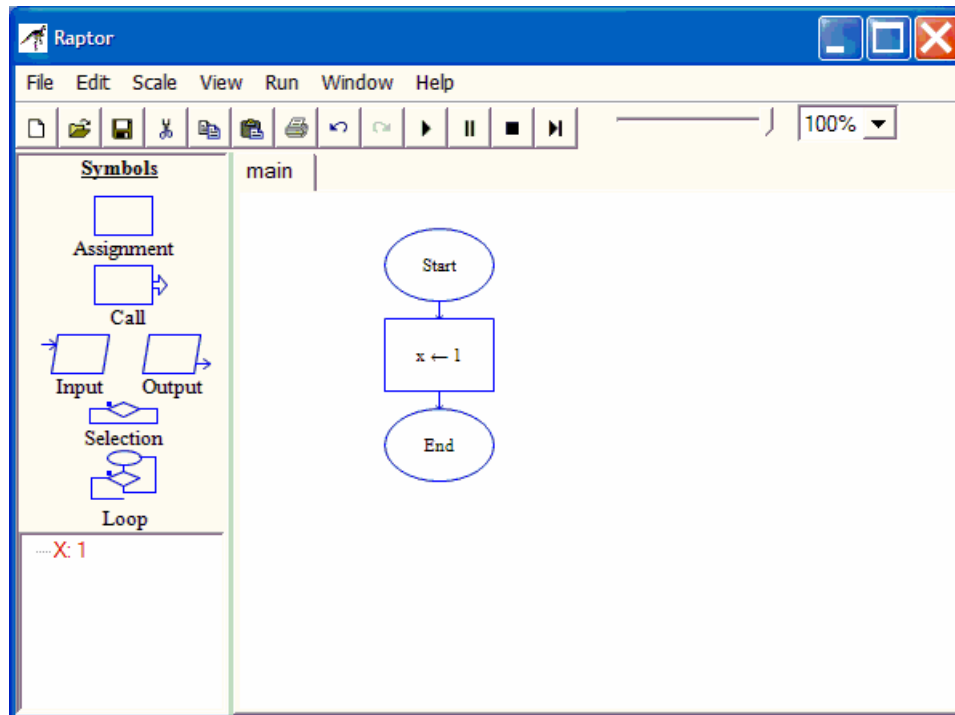


15-105 Raptor Tutorial

1. Raptor Overview

Raptor is a simple-to-use problem solving tool that enables the user to generate executable flowcharts. Raptor was written for students being introduced to the computing discipline in order to develop problem solving skills and improve algorithmic thinking.

The primary window consists of four main areas:



The Symbols area in the upper left presents the 6 primary graphical symbols that can be used in Raptor:

- The assignment symbol is used to give a variable a numeric or string value.
- The call symbol is used to make calls to outside procedures, such as graphics routines.
- The input symbol is used for getting input from the user.
- The output symbol is used to display text to the Master Console.
- The selection structure is used for decision making.
- The loop structure is used for iteration and repetition.


The area immediately below the Symbols area is the Watch Window. This area allows the user to view the current contents of any variables and arrays as the flowchart is executing.

The large, white area to the right is the primary Workspace. Users can build their flowcharts in this area and watch them update as they execute. The workspace is tabbed. Most flowcharts have a single tab named "main", but programmer-defined subcharts appear as tabs in the workspace.

The final area is the menu and toolbar. This area allows the user to change settings and control the view and execution of individual flowcharts.

2. Adding Symbols to 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.

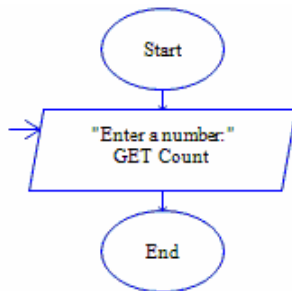
Once a symbol has been correctly placed, you can double click the symbol to edit its contents.

3. Simple Raptor Program

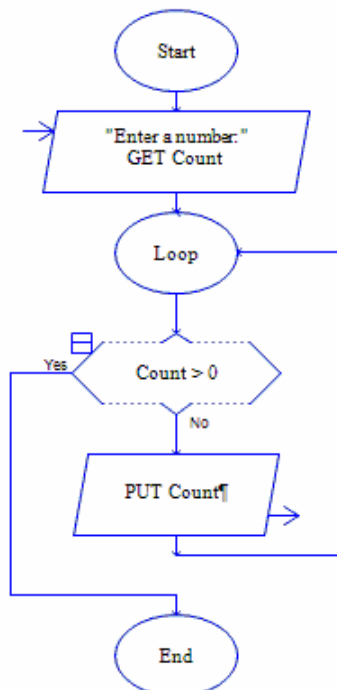
This brief tutorial will guide you through building and executing a simple flowchart.

We would like a flowchart that asks the user for a number, and then counts down to zero (displaying each number to the MasterConsole).

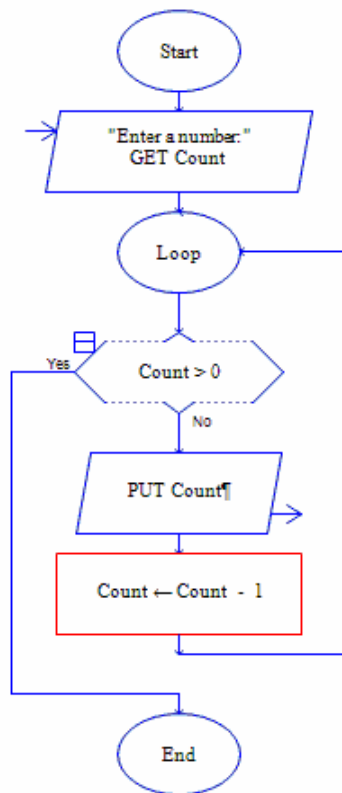
1. Start Raptor.
2. Add a Input symbol to the flowchart by left-clicking once on the Input symbol and then adding it between Start and End. Double click to add the prompt "Enter a number:" (including the quotes) and use the variable name **Count**.




3. Next, add a Loop structure and edit its expression to **Count < 0**
4. Add an Output symbol to output the variable Count to the MasterConsole.



5. Add an Assignment symbol to decrement the value of Count. Edit the statement to **Count = Count - 1**



6. To execute the flowchart, click on .

4. More Details About Raptor Symbols

The six symbols used in Raptor are displayed in the Symbol Window in the upper left corner of the main window.

4.1 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 " $x \leftarrow x + 1$ " 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.

4.2 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.

4.3 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. The user can also override the source for input by specifying a text file to be used in place of the keyboard.

4.4 Output

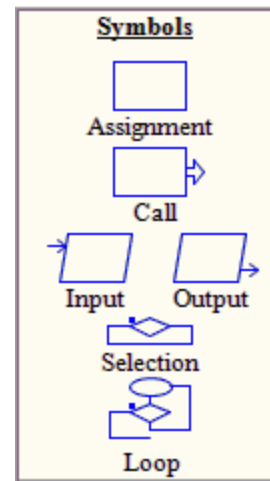
The output symbol is used to either write a number or text to the Master Console window. The user can also override the destination for output by specifying a text file to be used instead of the Master Console.

4.5 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).

4.6 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).



5. Glossary of Important Terms

Term	Notes
Variable	<p>A variable can be thought of as a name associated with a value. A variable is given a value in an Input symbol or in an Assignment symbol. The first time a variable is given a value, certain attributes for the variable are established:</p> <ul style="list-style-type: none">• Its type - does it hold a string or a number?• Its structure - does it hold a single value, or is it part of an array? <p>Once these attributes are established, they cannot change.</p>
String	<p>Strings are sequences of characters. A string variable holds a sequence of characters. Note that, once Raptor identifies a variable as a string variable, it cannot later be used to hold a number. Similarly, a numeric variable cannot later be used to hold a string. One last detail: Raptor doesn't support arrays of strings.</p>
Array	<p>An array is a collection of variables sharing the same name but accessed individually using an index. The index appears in square brackets ([]) after the array variable name. The number in the square brackets is the index. It distinguishes between the different variables falling under the name scores. Important Note: The index must be a positive integer; it cannot be 0 or negative, and it cannot have a fractional part.</p>
Procedure / Function	<p>Both procedures and functions have names. Both can have zero, one, two, or more arguments. If arguments are needed, they are listed in parentheses separated by commas after the name. For example, the Put_Pixel procedure requires three arguments to work—the x and y coordinates of the pixel to be set and the desired color to make the pixel. It is called in a Call symbol.</p>
Boolean Expressions	<p>The diamond symbol is used for decision making, either in a selection or loop structure. 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). In the case of a diamond controlling exit from a loop, a Yes or True value results in an exit from the loop, while a No or False value causes control to continue in the loop and eventually to the top of the loop again.</p>
Subchart	<p>Often, we want a program to perform the same steps at different places in the program. The simple solution is to copy and paste the desired steps from one place to another in the program. Unfortunately, this can enlarge our program, making it harder to work with. An alternative is to create a subchart that performs the desired steps, then call the subchart at the places in our program where those steps must be performed. Subcharts can break the Raptor program into logical parts. This simplifies design, ensures that flowcharts don't grow to unwieldy sizes, and reduces the chance for errors. Variables are shared throughout the main flowchart and all subcharts.</p>