

# Lab 07

## Problem Solving with Decision Structure

### Objective:

It will enable students to understand and use selection structures for data processing.

### Activity Outcomes:

The students will be able to:

- Solve problems related to selection control
- Solve problems related to nested selection control

### Instructor Note:

As a pre-lab activity, read “An Introduction to Programming and Algorithmic Reasoning using RAPTOR” by Hadfield, Weingart and Brown to gain an insight about flowchart representation in raptor.

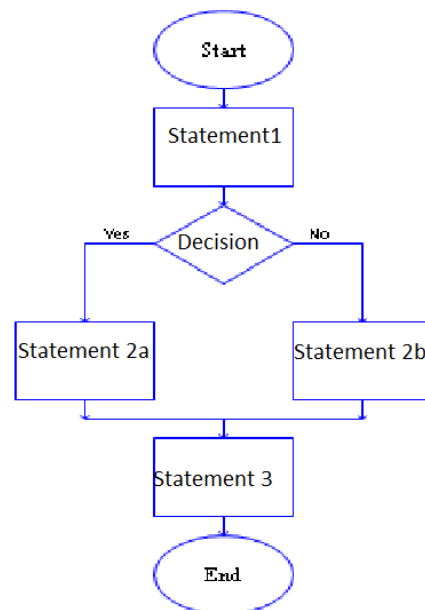
### 1) Useful Concepts

One of the most important parts of programming is controlling which statement will execute next. In this reading, you will learn the statements, called control structures or control statements that enable you, as a programmer, to determine the order in which your program statements are executed. Using these control structures, you can determine the order that statements are executed and whether or not statements are executed at all.

1. **if statement:** The *if* is a decision statement that allows the code to be executed based on a given boolean condition.

The algorithm you are developing may need to do some actions based upon a decision. For example, an algorithm that is evaluating a formula can check to see if a number is negative before taking the square root of that number and then proceed differently depending on whether the result will be a normal number or a complex number. Another example is processing grades. If the grade is above 90%, an A is assigned to the student, between 80% and 90%, a B is assigned to the student, and so on.

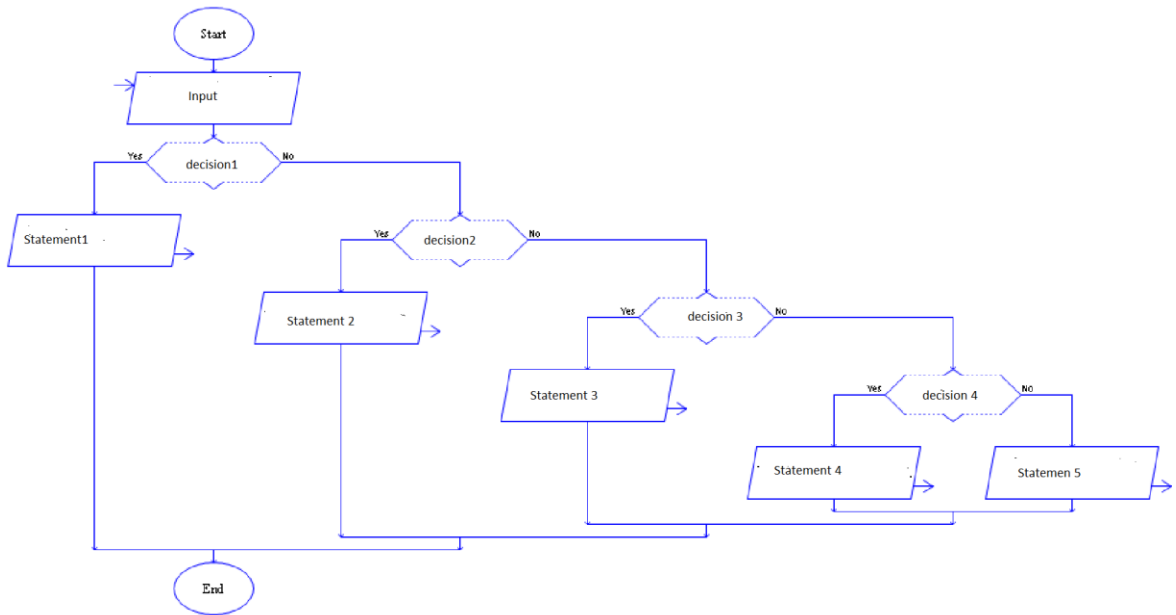
A selection-control statement controls whether or not a collection of code is executed or which collection of code is executed. In the inset diagram, the diamond shape represents a decision that when executed could result in an answer of Yes or No (True or False). Depending on what the answer is, the flow of control will follow the appropriate path. In the example, either statement 2a or statement 2b will be executed. One of them will be executed, but not both.



However, regardless of which statement 2 was executed, statement 3 will always be executed as it doesn't depend on any decision.

There need not be a statement 2a or a statement 2b. Either path could be empty or could contain several statements. It would be silly for both of them to be empty (or both have the exact same statements), as your decision, Yes or No, would have no effect (nothing different would happen based on the decision).

2. **Cascading Selection statements:** Sometime you are not making a selection between two alternatives, but making a decision amongst multiple alternatives. If this is the case, you need to have multiple selection statements.



## 2)Solved Lab Activites

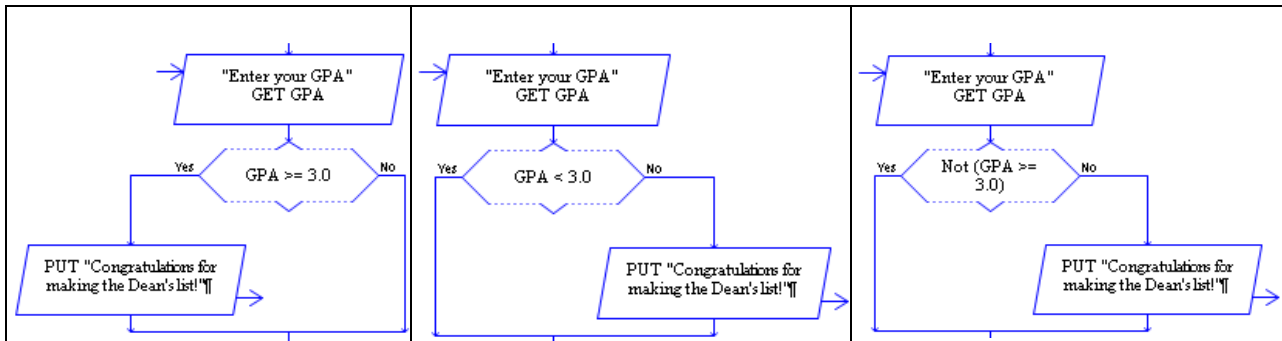
<i>Sr.No</i>	<i>Allocated Time</i>	<i>Level of Complexity</i>	<i>CLO Mapping</i>
<b>1</b>	<b>10</b>	<b>Medium</b>	<b>CLO-6</b>
<b>2</b>	<b>25</b>	<b>Medium</b>	<b>CLO-6</b>
<b>3</b>	<b>25</b>	<b>Medium</b>	<b>CLO-6</b>

## Activity 1:

*Raptor flowchart to illustrate decision structure*

### Solution:

The exact decision you write down affects whether the Yes or No paths are taken. It is very easy to change the decision to switch the yes and no paths. For example, on the left below is a simple decision on whether a student's GPA makes the grade for being on the Dean's list. On the right are two different rephrasings of the decision such that the No path is taken if the student is on the Dean's list. Some people prefer the version on the left and some the version in the center. Very few prefer the right because of the double negative implicit in the decision.

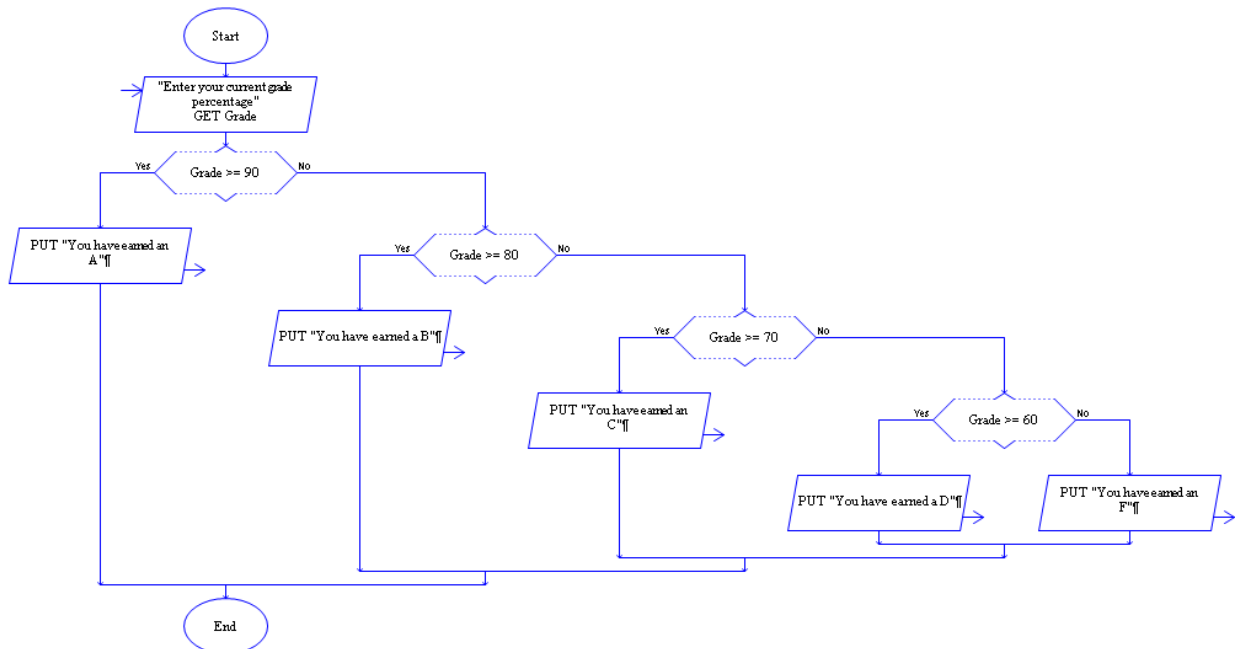


## Activity 2:

*Raptor flowchart to illustrate cascading decision structure*

### Solution:

Sometime you are not making a selection between two alternatives, but making a decision amongst multiple alternatives. For example, if you are assigning a grade (A, B, C, D, or F) you need to select

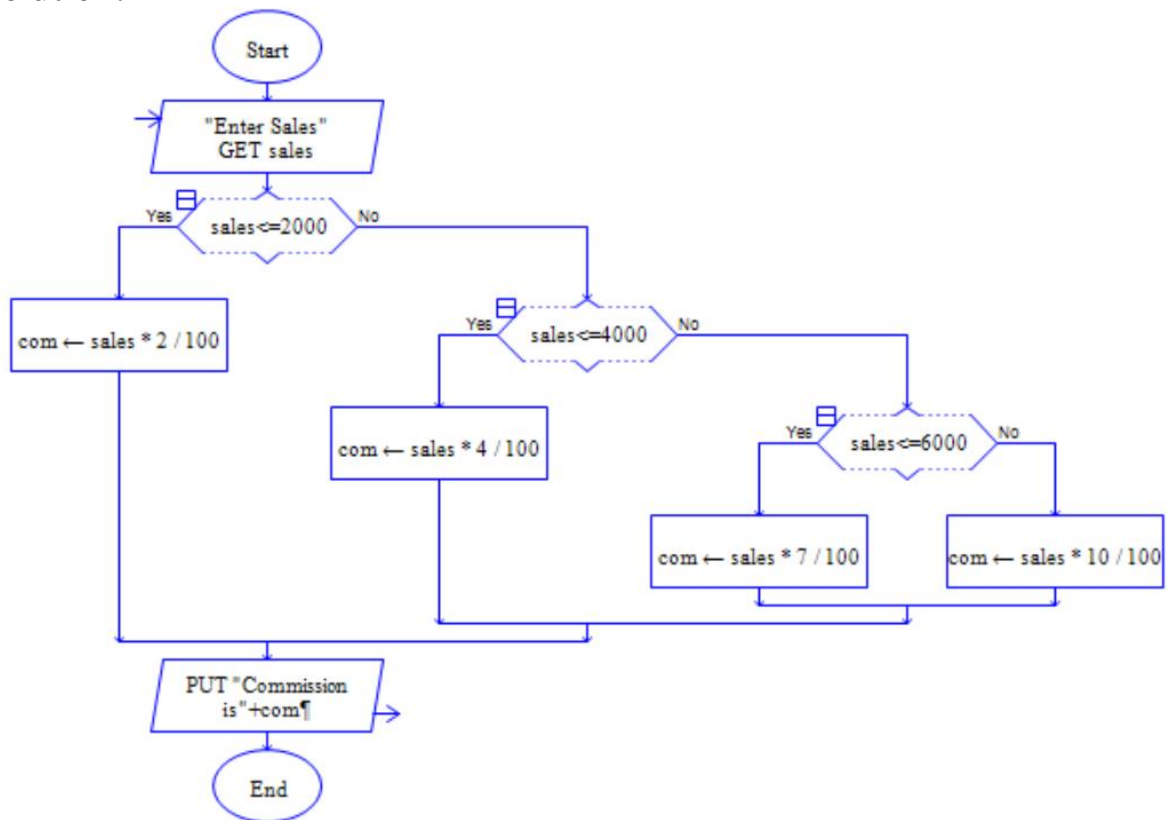


between the multiple choices. The following RAPTOR program includes multiple selection statements. If you follow the control logic, you can see how the selections cascade (as in water cascading over a series of falls).

### Activity 3:

*The problem illustrated in this flowchart is to calculate the commission rate for a salesperson, given the amount of sales. When the salesperson has sold less than or equal to \$2,000 worth of goods, the commission is 2%. When the sales total is more than \$2,000 and less than or equal to \$4,000, the commission is 4%. When the sales total is more than \$4,000 and less than or equal to \$6,000, the commission is 7%. When the person has sold more than \$6,000, the commission is 10%.*

#### Solution:



### 3. Graded Lab Tasks

*Note: The instructor can design graded lab activities according to the level of difficulty and complexity of the solved lab activities. The lab tasks assigned by the instructor should be evaluated in the same lab.*

#### Lab Task 1

Draw a flowchart to calculate the water bill given the cubic feet of water used for a Water Company, which charges the homeowner one of the following:

1. A flat rate of \$15.00 for usage up to and including 1000 cubic feet.
2. \$0.0175 per cubic foot for usage over 1000 cubic feet and up to and including 2000 cubic feet.
3. \$0.02 per cubic foot for usage over 2000 cubic feet and up to and including 3000 cubic feet.
4. A flat rate of \$70.00 for usage over 3000 cubic feet.

Test your flowchart with actual data.

#### Lab Task 2

A company that issues check-cashing cards uses an algorithm to create card numbers. The algorithm adds the digits of a four-digit number, and then adds a fifth digit of 0 or 1 to make the sum of the digits even. The last digit in the number is called the check digit. Draw a flowchart to develop a solution that accepts a four-digit number into one variable, adds the check digit, and prints the original number and the new number. Test your flowchart with the following data:

Original number = 4737      New number = 47371

And Original Number = 4631      New Number = 46310