Task Three: Decision Structures

In the last task, you outputted something based on what the user had inputted. However, this was simply copying what the user had inputted and output it.

Programmers often need the computer to make complicated decisions based on user input. That’s where decision structures come in. However, before learning about decision structures, it is necessary to understand a few common binary operators:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| + | - | \* | / | %\* |
| Addition | Subtraction | Multiplication | Division | Modulus (Remainder) |
| > | < | >= | <= | =\*\* |
| Greater Than | Less Than | Greater Than or Equal to | Less Than or Equal to | Equals |

\*Modulus is used to find the remainder. For example, 6 % 3 would equal 0, because 3 divides evenly into 6. However, 6 % 4 would equal 2, because 4 divides into 6 one time with 2 as a remainder.

\*\*The equals sign is used differently in java. It’s used to assign a value. You can say that int x = 2. That means that you are assigning a value of 2 to x.

These operators are the same as you are probably used to from math. However, when programming, it is often necessary to use other operators:

|  |  |  |
| --- | --- | --- |
| && | || | ! |
| And | Or | Not |

And can be used to check if both cases are true. For example, you could say:

If x && y, then do an action.

The action would happen only if x is true and if y is true.

Or is used to check if at least one of the cases is true. For example, you could say:

If x || y, then do an action.

The action would occur if x is true or if y is true, or if both of them are true.

Not is used to negate a value. If x is true, then !x is false.

These operators are often used in decision structures. Decision structures are used to check if a condition is met, and if so, proceed with an action.

if(condition)

{

action

}

A condition can be anything you would like to check. For example, you can check if one number is greater than another, or if a boolean is true or false.

Task:

Your task is to create a “choose your own adventure” game. If you aren’t sure what this is, there is an example here:

Your game can get user input from either a Scanner or a JOptionPane. You must have at least three different paths on each choice. For example, your first question can be:

Would you like to go 1. North, 2. South, 3. East, or 4. West? Then each option (1, 2, 3, 4) must give the user an additional three choices as the game progresses. If you are still confused about this, ask a volunteer.

Good luck, and remember to ask if you have any questions.