CS 111

Variables, if statements, if/else, Boolean operators

Models for Variable Declaration/Assignment

- DATA_TYPE VARIABLE_NAME; // Declaration
  - int x;
  - int x, y;
- VARIABLE_NAME = VALUE; // Assignment
  - x = 0;
  - x = 1, y = 2;
- DATA_TYPE VARIABLE_NAME = VALUE; // Initialization
  - int x = 0;
  - int x = 1, y = 2;

Models for output/input

- cout << THING; // Output
  - e.g. cout << VARIABLE_NAME; // Outputs value stored in variable
- cout << “Enter DATA_TYPE”; // Prompts user for input
- cin >> VARIABLE_NAME; // Reads input value into variable
  - cin >> x;
  - cin >> x >> y; // user will type value, press enter; type next value, press enter

Variable Types

- Integer – int
  - int year = 2020;
- Double – double
  - double price = 1.96;
- String – string
  - string city = “New York City”;
- Character – char
  - char star = ‘*’;
- Boolean (True/False) – bool
  - bool is_last = false;

Model for if

```
if (CONDITION) {
        STATEMENT(s);
}
```

- If the CONDITION is true, execute the STATEMENT
- If the CONDITION is not true, do not execute the STATEMENT
    - In other words, if CONDITION is false, just skip the following code block

Model for if/else

```
if (CONDITION) {
        STATEMENT(s);
} else {
        STATEMENT(s);
}
```

- If CONDITION is true, execute first STATEMENT(s)
- If CONDITION is false, execute second STATEMENT(s) after else

Model for if/else if/else

```
if (CONDITION) {
        STATEMENT(s);
} else if (CONDITION) {
        STATEMENT(s);
} else {
        STATEMENT(s);
}
```

- Use this when you have three or more branches for your decision tree
- Use additional else if (CONDITION) statements for each additional decision tree branch

Boolean operators

- Use when you want to test more than one condition for a single if statement
- And - &&
  - In an expression connected by &&, all elements must be true for the entire expression to evaluate as true
- Or - ||
  - In an expression connected by ||, if any element is true, the entire expression is evaluated as true
- Not - !
  - !(true) = false
  - !(false) = true

Examples of conditions (single and compound)

- Examples using count = 0, limit = 10, x = 12, y = 15
  - (count == 0) && (limit < 20)
  - (count > 5)
  - (limit > 20) || (count < 5)
  - !(count == 12)
  - (x > y)
  - (count < 10) && (x < y)
  - (y > x)
  - (limit < 20) || ((limit / x) > 7)
  - (limit >= 5) && (limit <= 10) // limit between 5 and 10 (not 5 <= limit <= 10)

Lab 4.1 Pseudocode

```
Function Main
    // Determines if three integers are entered in increasing order
    Declare three integer variables x, y, z
    Ask the user to enter three different numbers
    If x is less than y and y is less than z
        print “Increasing” to the monitor
    Else if x is greater than y and y is greater than z
        print “Decreasing” to the monitor
    Otherwise
        print “Neither”
```