**LaGuardia Community College – Last Update**

08

**Fall**

Part 4: C++ Selection Statements

Dr. Andi Toce

Lecture Notes for MAC 101 (Introduction to Computer Science)

Last updated / viewed: September 21, 15

Table of Contents

[1. *if* and *if-else* Statements 2](#_Toc414541839)

[2. Comparison Operators and Nested *if-else* Statements 4](#_Toc414541840)

[3. The *switch* Statement 5](#_Toc414541841)

[4. Logical Operators 6](#_Toc414541842)

# *if* and *if-else* Statements

Very often we need to make a decision whether to execute one or more statements based on criteria that need to be satisfied.

Most simple if – statement

**if (condition)**

**statement;**

|  |  |
| --- | --- |
| IfStatementExample.cpp | Output |
| #include <iostream>  using namespace std;  int main()  {  int x = 5;  int y = 5;  if (x==y)  cout << "x and y are equal \n";  cout << "Done!";  return 0;  } | x and y are equal  Done! |

Compound if – statement

**if (condition){**

**statement1;**

**statement2;**

**statement3;**

**}**

|  |  |
| --- | --- |
| CompoundIfExample.cpp | Output |
| #include <iostream>  using namespace std;  int main()  {  int x = 5;  int y = 5;  if (x == y){  cout << "x and y are equal \n";  cout << "x is " << x << " and y is " << y << endl;  }  cout << "Done!";  return 0;  } | x and y are equal  x is 5 and y is 5  Done! |

**Try now:** Write a program that inputs three integers from the keyboard and prints the largest value. Use the if-statement.

Example:

Input three different integers: 7 23 14

The largest value is: 23

**if – else statement.**

Statement1 is executed if condition is true. Statement2 if condition is false.

**if (condition)**

**statement1;**

**else**

**statement2;**

|  |  |
| --- | --- |
| IfElseExample.cpp | Output |
| #include <iostream>  using namespace std;  int main()  {  int x = 5;  int y = 6;  if (x == y)  cout << "x and y are equal \n";  else  cout << "x and y are not equal \n";  cout << "Done!";  return 0;  } | x and y are not equal  Done! |

**Try now:** Write a program that inputs an integer and prints if the number is even or odd.

Example:

Enter an integer: 24

The integer is even

# Comparison Operators and Nested *if-else* Statements

|  |  |  |  |
| --- | --- | --- | --- |
| Operator | Comparison | Operator | Comparison |
| == | Equal to | > | Greater Than |
| != | Not Equal to | <= | Less or Equal to |
| < | Less Than | >= | Greater or Equal to |

Example nested if-else statement

|  |  |
| --- | --- |
| ScoreToGrade.cpp | Output |
| #include <iostream>  using namespace std;  // Score to Grade Program  int main()  {  int score=0; //score on exam  cout << "Enter Score: ";  cin >> score;  cout << "The Grade is: ";  if ( score >= 90 ) // 90 and above gets "A"  cout << "A";  else if ( score >= 80 ) // 80-89 gets "B"  cout << "B";  else if ( score >= 70 ) // 70-79 gets "C"  cout << "C";  else if ( score >= 60 ) // 60-69 gets "D"  cout << "D";  else // less than 60 gets "F"  cout << "F";  return 0;  } | Enter Score: 88  The Grade is: B |

An alternative to if-else statement (using the ?: operator)

|  |  |
| --- | --- |
| PassFailScore.cpp | Output |
| #include <iostream>  using namespace std;  // Pass or Fail Program  int main()  {  int score=0; //score on exam  cout << "Enter Score: ";  cin >> score;  cout << ( score >= 60 ? "Passed" : "Failed" );  return 0;  } | Enter Score: 76  Passed |

# The *switch* Statement

The switch statement is an alternative to if-else. It is commonly used if we have several condition cases. It is easier to write and read.

Example switch statement compared to the equivalent if-else statement.

|  |  |
| --- | --- |
| SwitchWeekDay.cpp | IfElseWeekDay.cpp |
| #include <iostream>  using namespace std;  // Choose day of week program  int main()  {  int day = 0; // day of the week  cout << "Enter a number between 1 and 7 for the day of the week: ";  cin >> day;  switch (day)  {  case 1 : cout << "Monday";  break;  case 2 : cout << "Tuesday";  break;  case 3 : cout << "Wednesday";  break;  case 4 : cout << "Thursday";  break;  case 5 : cout << "Friday";  break;  case 6 : cout << "Saturday";  break;  case 7 : cout << "Sunday";  break;  default : cout << "Not a valid entry";  break;  }  return 0;  } | **#include** <iostream>  **using** **namespace** std;  // Choose day of week program  **int** **main**()  {  **int** day = 0; // day of the week  cout << "Enter a number between 1 and 7 for the day of the week: ";  cin >> day;  **if** (day == 1) cout << "Monday";  **else** **if** (day == 2) cout << "Tuesday";  **else** **if** (day == 3) cout << "Wednesday";  **else** **if** (day == 4) cout << "Thursday";  **else** **if** (day == 5) cout << "Friday";  **else** **if** (day == 6) cout << "Saturday";  **else** **if** (day == 7) cout << "Sunday";  **else** cout << "Not a valid entry";  **return** 0;  } |

**Try now:** Write a program that inputs a number between 1 and 12 and prints the corresponding month of the year. If a different value is entered the program prints: “Not a valid entry!”

Example:

Enter a number between 1 and 12 for the month of the year: 2

The month you selected is: February

# Logical Operators

|  |  |  |
| --- | --- | --- |
| Operator | Logical Operation | Example |
| ! | NOT | !(2==2) // evaluates to FALSE |
| && | AND | (2==2) && (2==3) //evaluates to FALSE |
| || | OR | (2==2) || (2==3) //evaluates to TRUE |

The logical operator && yields TRUE only if both conditions are TRUE.

The logical operator || yields TRUE if either condition is TRUE.

**Logical Tables**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | && Operator (AND) | | | | A | B | A && B | | TRUE | TRUE | TRUE | | TRUE | FALSE | FALSE | | FALSE | TRUE | FALSE | | FALSE | FALSE | FALSE | | |  |  |  | | --- | --- | --- | | || Operator (OR) | | | | A | B | A || B | | TRUE | TRUE | TRUE | | TRUE | FALSE | TRUE | | FALSE | TRUE | TRUE | | FALSE | FALSE | FALSE | |

**OR (||) operator example**

|  |  |
| --- | --- |
| OrOperatorExample.cpp | Output |
| #include <iostream>  using namespace std;  int main ()  {  char agree;  cout << "Would you like to meet me? (y/n): ";  cin >> agree;  if (agree == 'y' || agree == 'Y')  cout << "Great! Looking forward to meeting you!"<< endl;  else if (agree == 'n' || agree == 'N')  cout << "Sorry to hear that! Good luck!" << endl;  else  cout << "Please don't play games! Tell me yes or no!" << endl;  return 0;  } | Would you like to meet me? (y/n): Y  Great! Looking forward to meeting you! |

**AND (&&) operator example**

|  |  |
| --- | --- |
| AndOperatorExample.cpp | Output |
| #include <iostream>  using namespace std;  int main ()  {  int number;  cout << "I only like numbers between 10 and 20. Enter an integer: ";  cin >> number;  if (number >= 10 && number <= 20)  cout << "You entered "<< number << ". I am happy!" << endl;  else  cout << "I don't like the number you entered!" << endl;  return 0;  } | I only like numbers between 10 and 20. Enter an integer: 15  You entered 15. I am happy! |

**Try Now:** Determine whether the following expressions are true or false. Use a computer program to verify your answers.

1. (4 == 4) && (4 >= 3) =
2. (4 == 4) || (10 < 5) =
3. !(4 == 4) || (10 < 5) =
4. !(4 == 4) || !(10 < 5) =
5. (3 != 4) && (10 < 15) || (4 == 5) =
6. (3 != 4) && (10 < 15) && (4 == 5) =