

[Ravi Patel]

Instructor: Dr. Thamira Hindo

[CPSC 230]

Chapter 2- lab assignment.... If statement and while loops

Note: Submit your assignment in the inbox (chapter 2 assignment).

Part 2 (15 pts)

Due date: End of the class time

Q1- (7 pts)

- 1- Write an if-else statement that outputs the word High if the value of the variable score is greater than 100 and Low if the value of score is at most 100. The variable score is of type int.

```
if (score > 100)
cout << "High";
else
cout<<"Low";
```

- 2- Write an if-else statement that outputs the word Warning if the temperature is greater than or equal to 100, or the pressure is greater than or equal to 200, or both. Otherwise, the if-else statement outputs the word OK. The variables temperature and pressure are both of type int.

```
if((temperature >=100) || (pressure) >=200)
cout<<"Warning";
else
cout<<"OK";
```

- 3- What is the output produced by the following (when embedded in a correct program with x declared to be of type int)?

```
    x = 10;
    while (x > 0)
{
    cout << x << endl;
    x = x - 3;
}
```

```
//OUTPUT:
//          10
//          7
//          4
//          1
```

Q 2- (3 pts.)

The velocity of sound in dry air can be approximated by the formula:

$$\text{Velocity} = 331.3 + 0.61 * T_c$$

Where T_c is the temperature of the air in degrees Celsius and the velocity is in m/sec. Write a program to find the velocity when the temperature = 10:10:50 in the following format:

At 10 degrees Celsius the velocity of sound is ----- m/s

At 20 degrees Celsius the velocity of sound is ----- m/s

At 30 degrees Celsius the velocity of sound is ----- m/s

At 40 degrees Celsius the velocity of sound is ----- m/s

At 50 degrees Celsius the velocity of sound is ----- m/s

```
#include <iostream>
using namespace std;
int main(int argc, char *argv[]) {

double temp1, temp2;
double velocity;
char n;

do {
cout << "Desired Starting Temperature: "; //ask for user input
cin >> temp1; //assign input to temp1 (starting temperature)
cout << "Desired Ending Temperature: "; //ask for user input
cin >> temp2; //assign input to temp2 (ending temperature)

cout << "\n";

while (temp1 <= temp2) { //while the starting temp is less than or equal to the
ending temp

velocity = 331.3 + 0.61 * temp1; //calculate velocity
cout << "At " << temp1 << " degrees Celsius the velocity of sound is " << velocity
<< " m/s" << endl;
temp1 += 10; //add 10 to temp1 each time the do-while loop is run

}

cout << "\nAny more temperatures to input?"; //ask user if they want to continue
program
cout << "\nEnter 'y' to continue || 'n' to stop the program: ";
cin >> n; //if n is entered, stop program
cout << "\n";

}

while (n == 'y' || n == 'Y'); // end do-whileloop unless y or Y is entered
```

```

}

//OUTPUT:
//Desired Starting Temperature: 10
//Desired Ending Temperature: 50

//At 10 degrees Celsius the velocity of sound is 337.4 m/s
//At 20 degrees Celsius the velocity of sound is 343.5 m/s
//At 30 degrees Celsius the velocity of sound is 349.6 m/s
//At 40 degrees Celsius the velocity of sound is 355.7 m/s
//At 50 degrees Celsius the velocity of sound is 361.8 m/s

//Any more temperatures to input?
//Enter 'y' to continue || 'n' to stop the program: n

```

Q 3- (5 pts.)

The Babylonian algorithm to compute the square root of a number n is as follows:

- Make a guess at the answer (may be $n/2$)
- Compute $r=n/\text{guess}$
- Set $\text{Guess}=(\text{guess} + r)/2$
- Go back to step b for as many iterations as necessary. The more that steps b and c repeats, the closer guess will become the square root of n .

```

#include <iostream>

using namespace std;
int main(int argc, char *argv[]) {

double n;
    cout<<"What is n? (number to find sqrt of): ";
    cin>>n;
    cout<<"\n";
double guess = n / 2; // calc guess for n
double r = n / guess; // calc r for n

while(guess*guess < 0.99*n || guess*guess > 1.01*n) { //guess * guess == sqrt n in
theory, we use 0.99 and 1.01 to check if guess * guess is close enough to sqrt n,
because it may not be exactly the same (due to continuous decimals)
    guess = (guess + r) / 2; //run the while loop again, adding guess + r &
dividing by 2
    r = n / guess;

    } // end while loop

    cout << "Babylonian square root guess = "<< guess; //display guess, proven
that it is corrected enough at this point

}

//SAMPLE OUTPUT:
//What is n? (number to find sqrt of): 16

//Babylonian square root guess = 4.00122

```