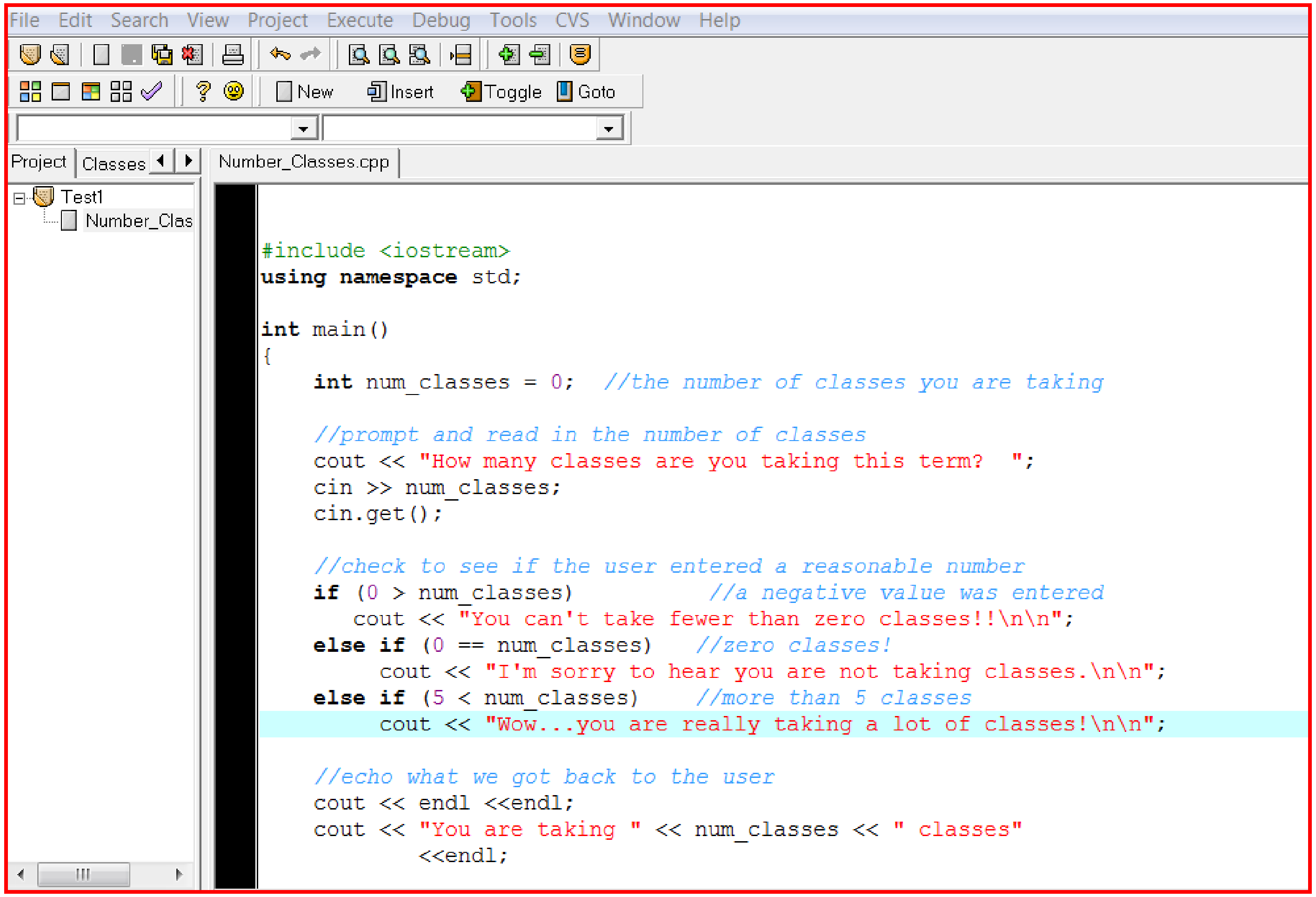
Today in CS161

- ***Week #4***
  - ***Using if statements***
    - **Create new Programs**
      - Using if and else statements
    - **Create new Graphics Demos**
      - Using if and else statements
  - ***Learn about Repetition***
    - **Loops!**

Using If statements in a program

- We will now do two examples, one extending the program we wrote last week finding out how many classes you are taking (non-graphical
- And…another that takes the graphics program and finds out if the user wants to draw circles or rectangles!
- It is all about choices!
- Plus, you will notice that now we can error check

Integrating this into a program…

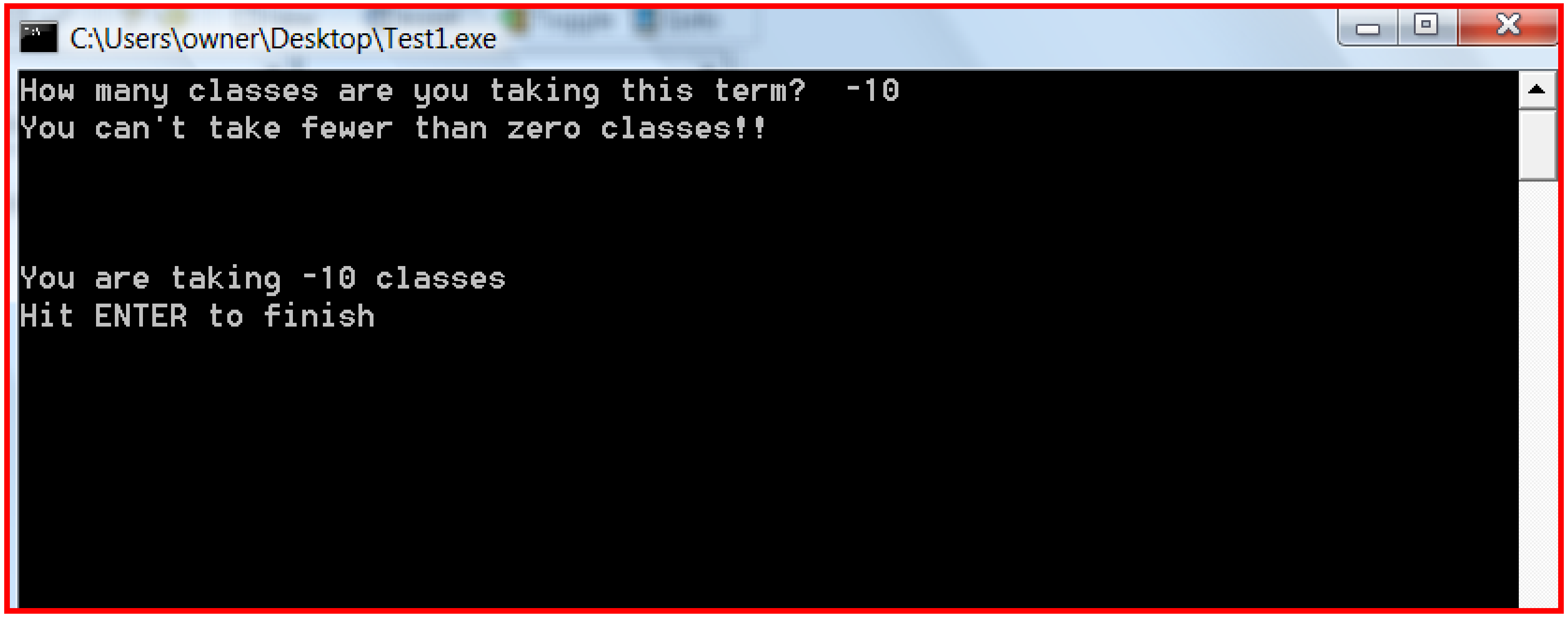


```
#include <iostream>
using namespace std;

int main()
{
    int num_classes = 0;  //the number of classes you are taking

    //prompt and read in the number of classes
    cout << "How many classes are you taking this term?  ";
    cin >> num_classes;
    cin.get();

    //check to see if the user entered a reasonable number
    if (0 > num_classes)             //a negative value was entered
       cout << "You can't take fewer than zero classes!!\n\n";
    else if (0 == num_classes)    //zero classes!
         cout << "I'm sorry to hear you are not taking classes.\n\n";
    else if (5 < num_classes)     //more than 5 classes
         cout << "Wow...you are really taking a lot of classes!\n\n";

    //echo what we got back to the user
    cout << endl <<endl;
    cout << "You are taking " << num_classes << " classes"
            <<endl;
```

Integrating this into a program…

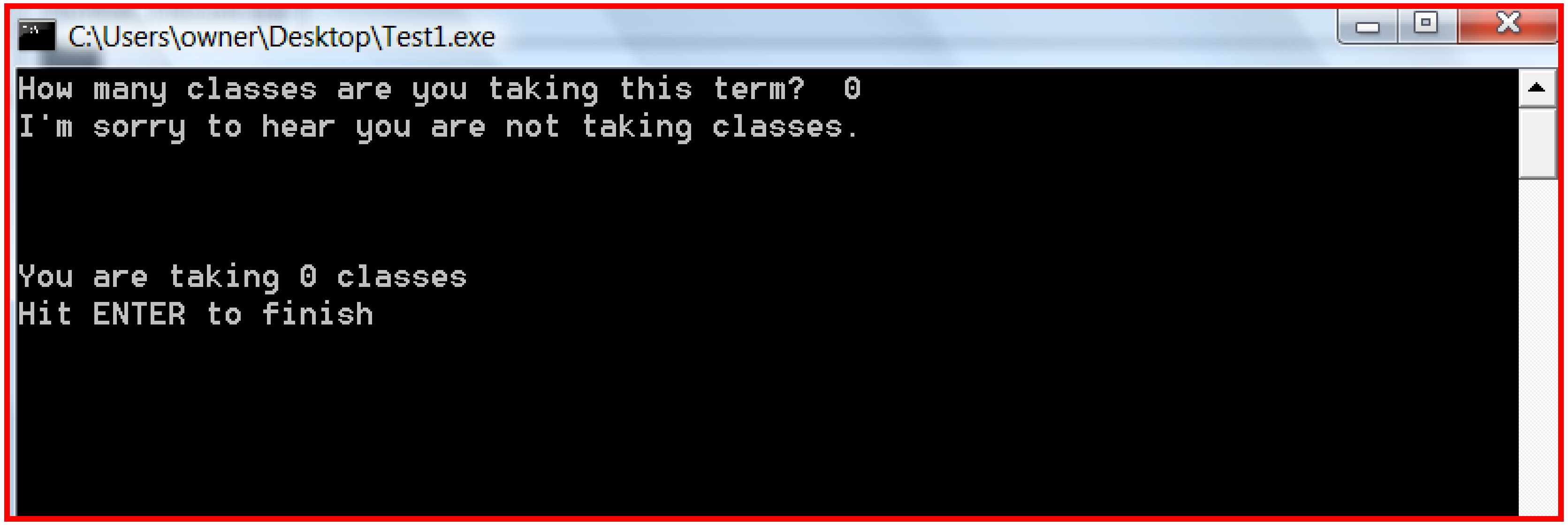
```
int num_classes = 0;  //the number of classes you are taking

//prompt and read in the number of classes
cout << "How many classes are you taking this term?  ";
cin >> num_classes;
cin.get();


//check to see if the user entered a reasonable number
if (0 > num_classes)          //a negative value was entered
    cout << "You can't take fewer than zero classes!!\n\n";
else if (0 == num_classes)   //zero classes!
    cout << "I'm sorry to hear you are not taking classes.\n\n";
else if (5 < num_classes)    //more than 5 classes
    cout << "Wow...you are really taking a lot of classes!\n\n";
```

Integrating this into a program…



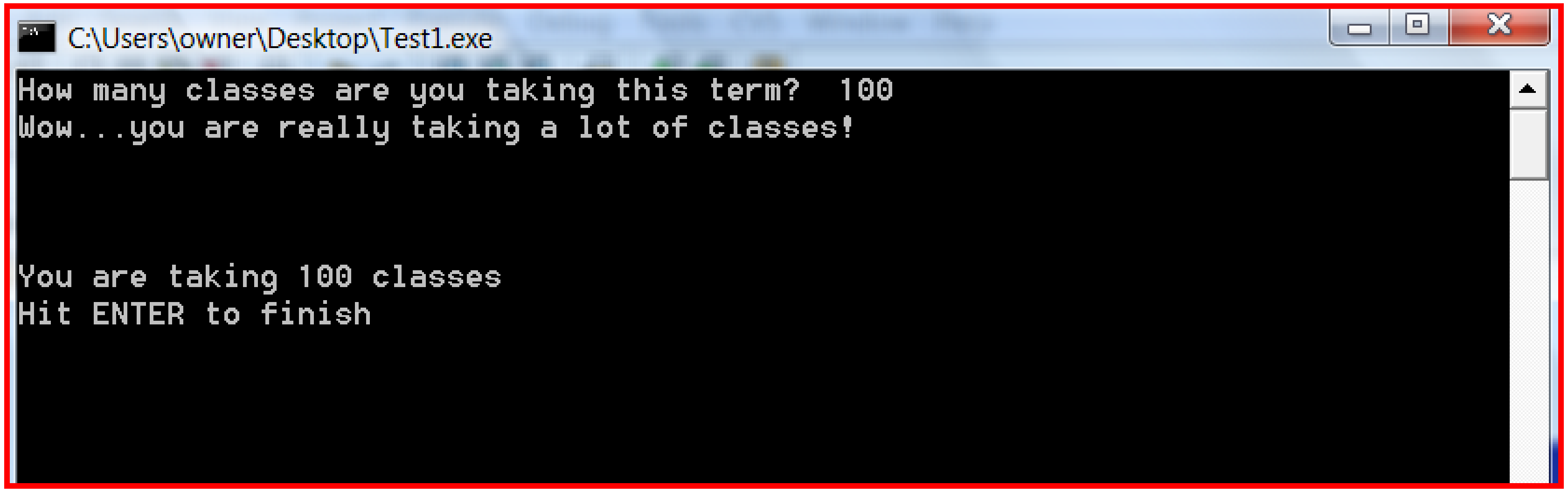
```
How many classes are you taking this term?  -10
You can't take fewer than zero classes!!



You are taking -10 classes
Hit ENTER to finish
```

Integrating this into a program…


C:\Users\owner\Desktop\Test1.exe

```
How many classes are you taking this term?  0
I'm sorry to hear you are not taking classes.



You are taking 0 classes
Hit ENTER to finish
```

Integrating this into a program…



```
How many classes are you taking this term?  100
Wow...you are really taking a lot of classes!



You are taking 100 classes
Hit ENTER to finish
```

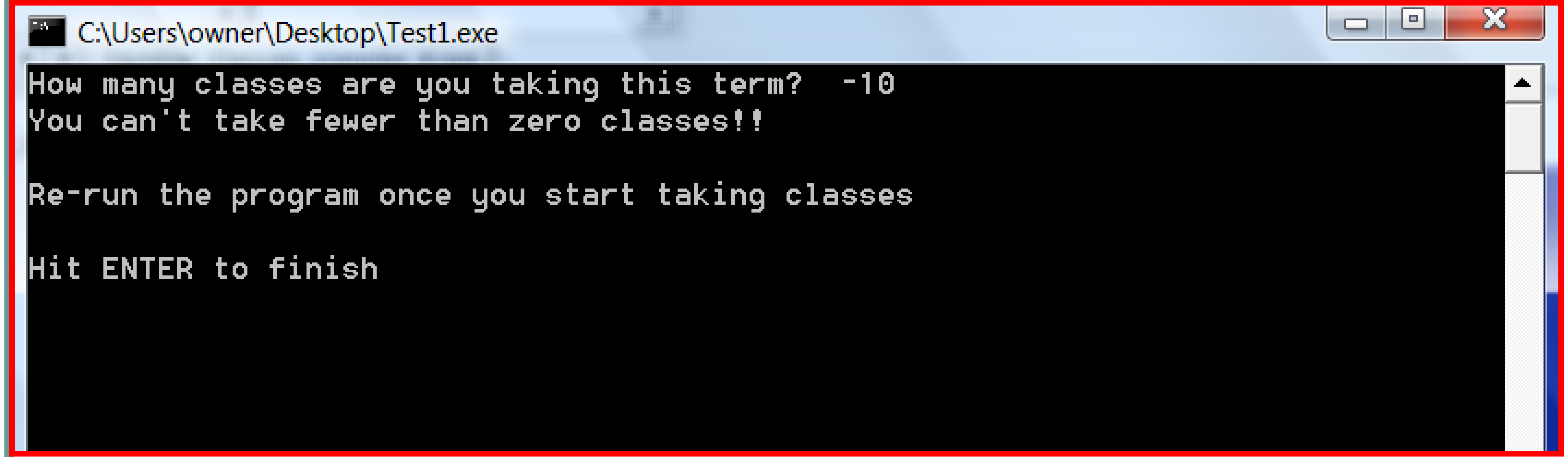
Using If statements in a program

- Let’s extend this now to only echo the number of classes that you are taking, if it is a reasonable number

  - Otherwise, after the error message we will tell the user to re-run the program once they have a better idea of how many classes they are taking

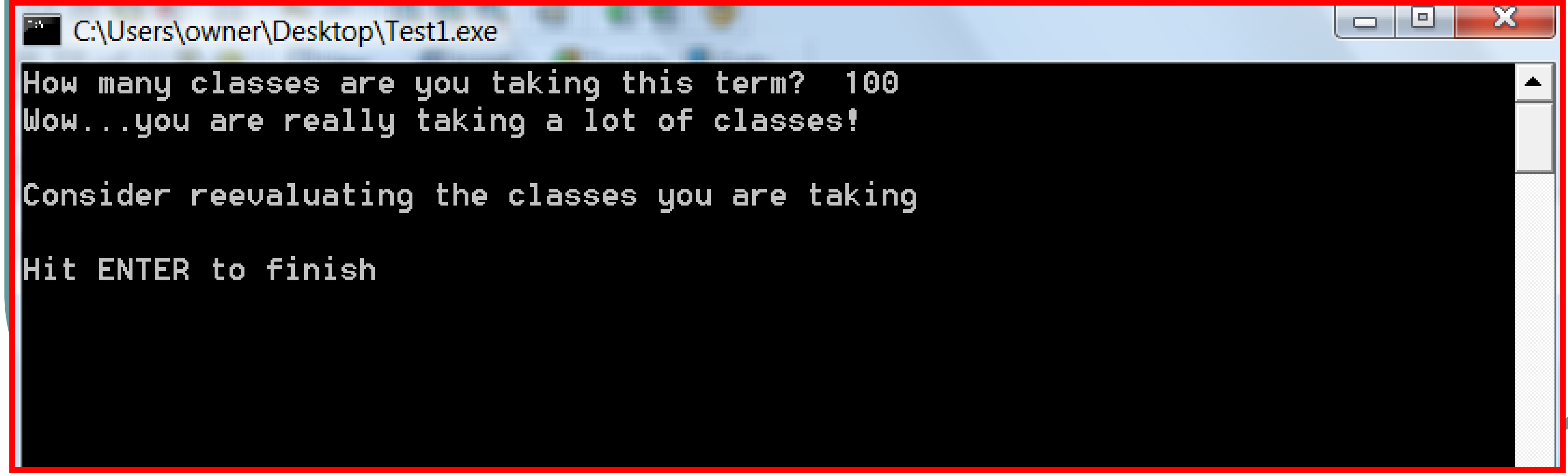
```
//check to see if the user entered a reasonable number
    if (0 > num_classes)          //a negative value was entered
       cout << "You can't take fewer than zero classes!!\n\n";
    else if (0 == num_classes)   //zero classes!
         cout << "I'm sorry to hear you are not taking classes.\n\n";
    else if (5 < num_classes)    //more than 5 classes
         cout << "Wow...you are really taking a lot of classes!\n\n";
    else
    {

        //echo what we got back to the user, if it is valid
        cout << endl <<endl;
        cout << "You are taking " << num_classes << " classes"
              <<endl;
     }

     //tell the user to re-use the program if it was an invalid value
     if (num_classes <= 0)
        cout << "Re-run the program once you start taking classes\n\n";
     else if (num_classes > 5)
          cout << "Consider reevaluating the classes you are taking\n\n";
```

Running this new version...



C:\Users\owner\Desktop\Test1.exe

```
How many classes are you taking this term?  -10
You can't take fewer than zero classes!!

Re-run the program once you start taking classes

Hit ENTER to finish
```



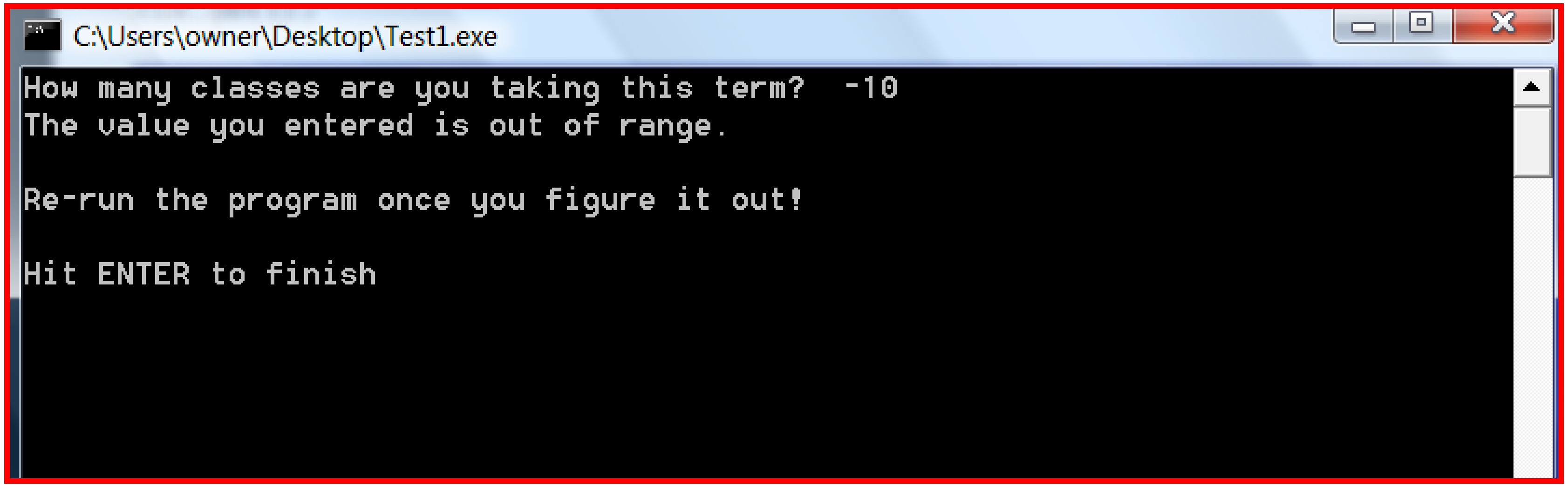
C:\Users\owner\Desktop\Test1.exe

```
How many classes are you taking this term?  100
Wow...you are really taking a lot of classes!

Consider reevaluating the classes you are taking

Hit ENTER to finish
```

Another approach...for next time

```
//check to see if the user entered a reasonable number
//If the number is less than or equal to zero OR greater than 5
if (0 >= num_classes || 5 < num_classes)     //out of range!
{
   cout << "The value you entered is out of range.\n\n";

   cout << "Re-run the program once you figure it out!\n\n";
}
else
{

    //echo what we got back to the user
    cout << endl <<endl;
    cout << "You are taking " << num_classes << " classes"
            <<endl;
 }
```

Running this new version…

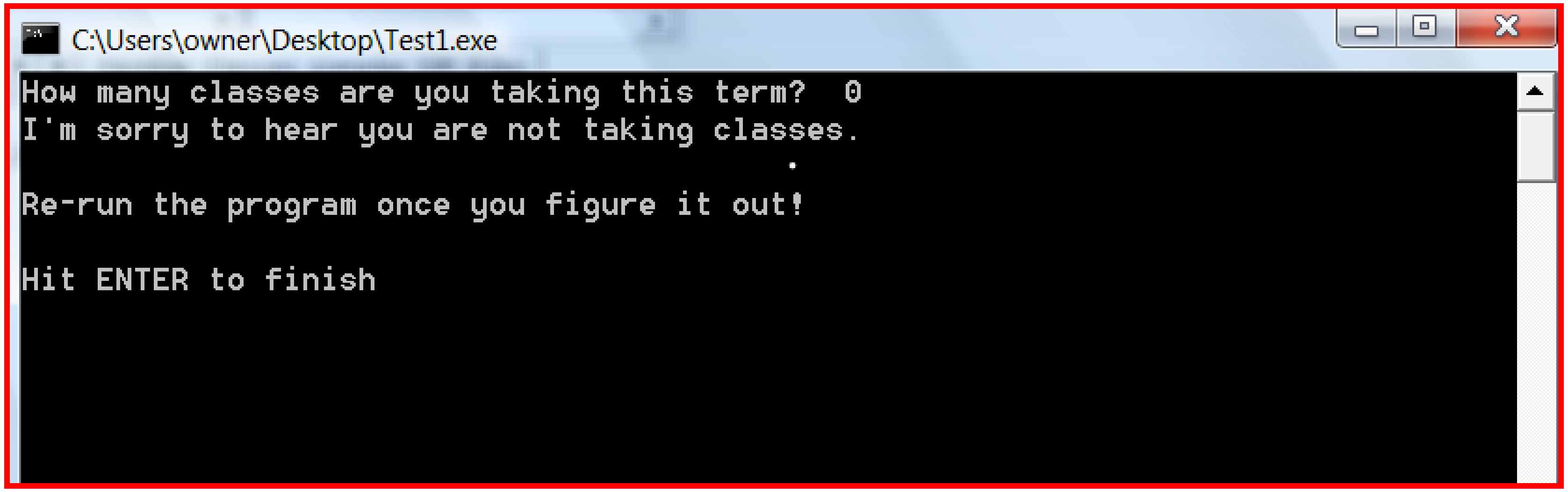


```
How many classes are you taking this term?  -10
The value you entered is out of range.

Re-run the program once you figure it out!

Hit ENTER to finish
```

Another approach...for next time

```
//check to see if the user entered a reasonable number
    if (0 >= num_classes || 5 < num_classes)     //out of range!
    {
       if (0 > num_classes)
          cout << "You can't take fewer than zero classes!!\n\n";
       else if (0 == num_classes)   //zero classes!
          cout << "I'm sorry to hear you are not taking classes.\n\n";
       else if (5 < num_classes)    //more than 5 classes
         cout << "Wow...you are really taking a lot of classes!\n\n";

       cout << "Re-run the program once you figure it out!\n\n";
    }
    else
    {
        //echo what we got back to the user
        cout << endl <<endl;
        cout << "You are taking " << num_classes << " classes"
              <<endl;
     }
```

Running this new version…



```
How many classes are you taking this term?  0
I'm sorry to hear you are not taking classes.
                                          .
Re-run the program once you figure it out!

Hit ENTER to finish
```

Using Graphics…

Let’s extend the graphics program from last time, asking the user if they want a Rectangle (R) or a Circle (C)

```
//Here is where I am going to put my variables
    int window_size;
    int color;
    int circle_radius;
    char selection;      // what does the user want to do?
    int width, height;  //rectangle width and height

    cout << "How big of a window do you want (pick a number less than 1200): ";
    cin >> window_size;   cin.get();
    initwindow(window_size, window_size);

    cout << "What color do you want...enter in a number 1-15 (15 is white) ";
    cin >> color;  cin.get();
    setcolor(color);
    setfillstyle(SOLID_FILL,color);
```

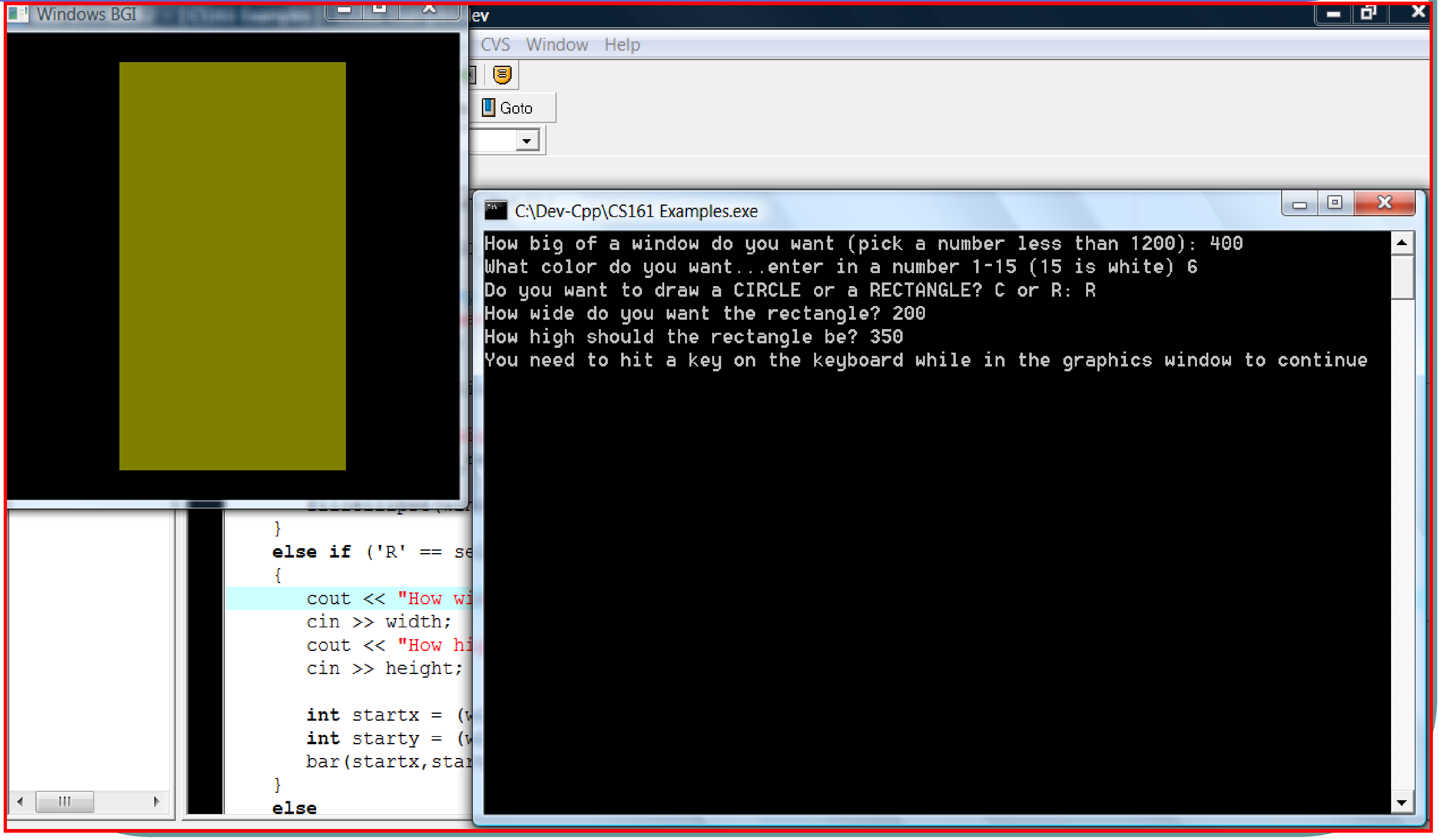
Using Graphics…

```
//Find out if they want to draw a circle or a rectangle
    cout << "Do you want to draw a CIRCLE or a RECTANGLE? C or R: ";
    cin >> selection;   cin.get();

    if ('C' == selection) //Circle
    {
       cout << "How big do you want the circle? ";
       cin >> circle_radius;   cin.get();

       fillellipse(window_size/2,window_size/2,circle_radius,circle_radius);
    }
    else if ('R' == selection)  //Rectangle
    {
       cout << "How wide do you want the rectangle? ";
       cin >> width;     cin.get();
       cout << "How high should the rectangle be? ";
       cin >> height;    cin.get();

     //  next page……
```

Using Graphics…

```
   //figure out how to draw the filled rectangle (a bar)
   int startx = (window_size-width)/2;   //center the rectangle
   int starty = (window_size-height)/2;
   bar(startx,starty,startx+width,starty+height); //"bars" are filled
}
else
{   //The user did not enter a C or an R
     cout <<"Sorry you couldn't decide!" <<endl <<endl;
     settextstyle(0,0,6); //6 is BIG

     outtextxy(0,window_size/2,"TOO BAD!");
}
```

Running this new version…



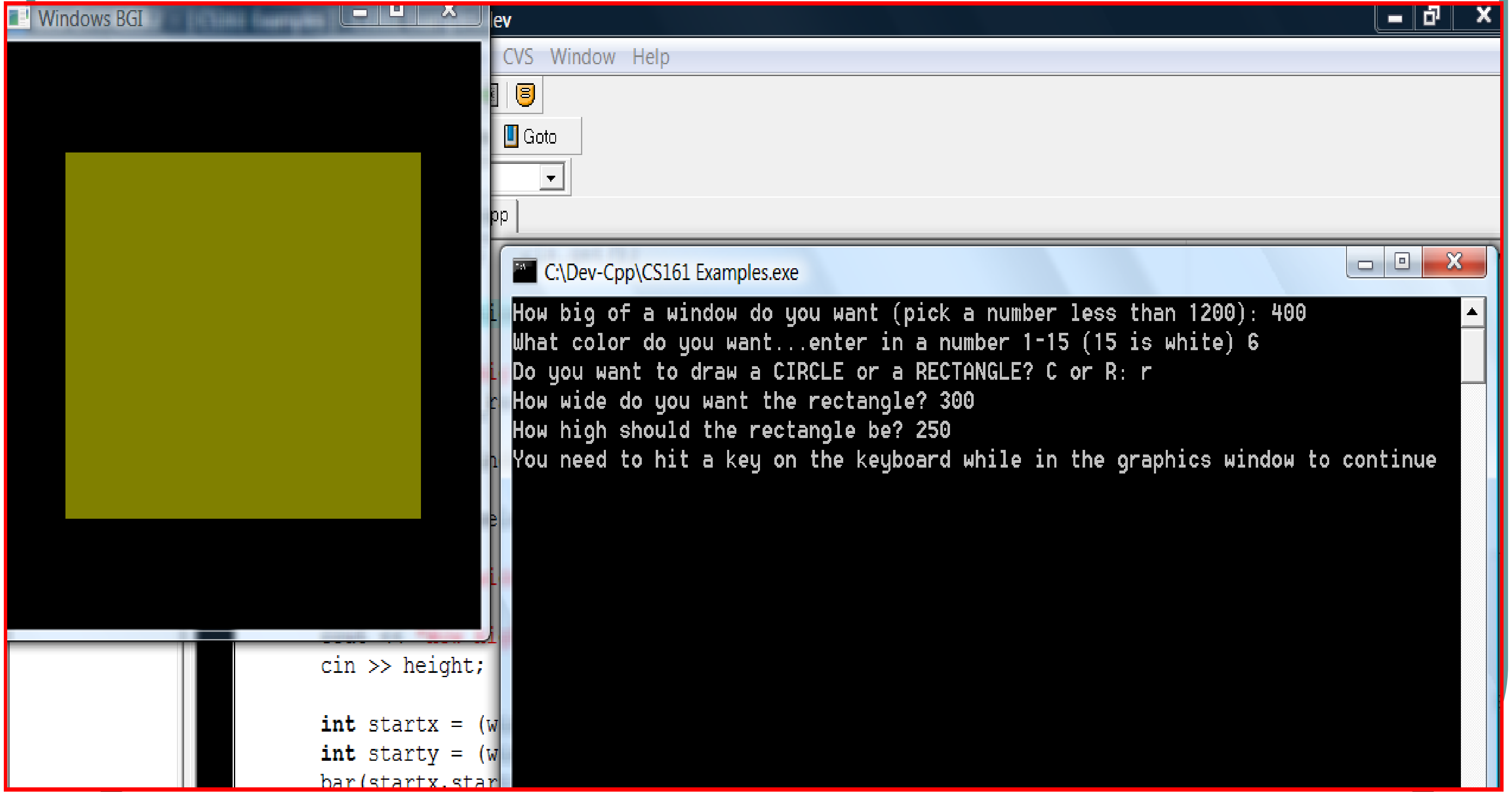
```
How big of a window do you want (pick a number less than 1200): 400
What color do you want...enter in a number 1-15 (15 is white) 6
Do you want to draw a CIRCLE or a RECTANGLE? C or R: R
How wide do you want the rectangle? 200
How high should the rectangle be? 350
You need to hit a key on the keyboard while in the graphics window to continue
```

```
    }
    else if ('R' == se
    {
        cout << "How wi
        cin >> width;
        cout << "How hi
        cin >> height;

        int startx = (w
        int starty = (w
        bar(startx,star
    }
    else
```

Using Graphics…using the OR

```
//Find out if they want to draw a circle or a rectangle
   cout << "Do you want to draw a CIRCLE or a RECTANGLE? C or R: ";
   cin >> selection;   cin.get();

   if ('C' == selection || 'c' == selection) //Circle (upper or lower case)
   {
     cout << "How big do you want the circle? ";
     cin >> circle_radius;   cin.get();

     fillellipse(window_size/2,window_size/2,circle_radius,circle_radius);
   }
   else if ('R' == selection|| 'r' == selection)  //Rectangle
   {
     cout << "How wide do you want the rectangle? ";
     cin >> width;     cin.get();
     cout << "How high should the rectangle be? ";
     cin >> height;    cin.get();

    //  ……etc. … no other changes
```

Running this new version…



```
How big of a window do you want (pick a number less than 1200): 400
What color do you want...enter in a number 1-15 (15 is white) 6
Do you want to draw a CIRCLE or a RECTANGLE? C or R: r
How wide do you want the rectangle? 300
How high should the rectangle be? 250
You need to hit a key on the keyboard while in the graphics window to continue
```

```
cin >> height;

int startx = (w
int starty = (w
bar(startx,star
```

Today in CS161

- ***Next Topic:***
  - ***Learn about…***
    - ***Logicals: And, Or, Not***
    - ***Repetition: loops!***
  - **Rewrite our First Program**
    - Using loops
  - **Graphics**
    - Let's make the graphics move….

Logical Operators

- **There are 3 logical (boolean) operators:**

&&	**And (operates on two operands)**
||	**Or (operates on two operands)**
!	**Not (operates on a single operand)**

- **&& evaluates to true if both of its operands are true;**
  - otherwise it is false.

Logical Operators

- **|| evaluates to true if one or the other of its operands are true;**
  - it evaluates to false only if both of its operands are false.
- **! gives the boolean complement of the operand.**
  - If the operand was true, it results in false.

Logical Operators

Conditional Expression	Logical value	True/False
(5 == 10) && (30 < 88)	0	False
(5 == 10) || (30 < 88)	1	True
!(5==10) && (30 < 88)	1	True
40 != 44	1	True
!(40 != 44)	0	False

AND Truth Table

- **op1 && op2 results in:**

op1	op2	residual value	
true	true	true	1
true	false	false	0
false	true	false	0
false	false	false	0

OR Truth Table

- **op1 || op2 results in:**

op1	op2	residual value	
true	true	true	1
true	false	true	1
false	true	true	1
false	false	false	0

NOT Truth Table

- **!op1 results in:**

op1	residual value	
true	false	0
false	true	1

Logicals in `if` Statements

- Now let’s apply this to the if statements.
- For example, to check if our input is only an ‘m’ or an ‘i’

```
char selection;
cin >> selection
if ((‘m’ != selection ) &&
   (‘i’ != selection) )
   cout << “Error! Try again”;
```

Logicals in `if` Statements

- **Why would the following be incorrect?**

```
if (('m' != selection ) ||
    ('i' != selection) )
    cout << "Error! Try again";
```

- Because no matter what you type in (m, i, p, q) it will <u>never</u> be both an m <u>and</u> an i!
- If an m is entered, it won't be an i!!!!!

Logicals in `if` Statements

- **Let’s change this to check if they entered in either an m or an i: (this is <u>correct</u>)**

```
if ((‘m’ == selection ) ||
   (‘i’ == selection ) )
   cout << “Correct!”;
else
      cout << “Error. Try Again!”;
```

Logicals in `if` Statements

- **Now, let’s slightly change this....**

```
if (!((‘m’ == selection ) ||
      (‘i’ == selection )))
   cout << “Error. Try Again!”;
```

- Notice the parens...you must have a set of parens around the logical expression

Repetition in Programs

- **What if we wanted to give the user another chance to enter their input...**
- This would be impossible without loops
- **Algorithms that require loops look something like:**
  - Step 1: Receive Input
  - Step 2: Echo the Input
  - Step 3: Ask the user if this is correct
  - Step 4: If not, <u>repeat</u> beginning at step #1

Three types of Loops

- **There are three ways to repeat a set of code using loops:**
  - while loop
  - do while loop
  - for loop
- Each of these can perform the same operations...
  - it is all in how you think about it!

....let’s see....

Using a While Loop

- Let's give the user a 2nd (and 3rd, 4th, 5th...) chance to enter their data using a while loop.
- While loops have the form: (notice semicolons!)

```
while (logical expression)
          single statement;
```

```
while (logical expression)
{
          many statements;
}
```

Using a While Loop

- The while statement means that while an expression is true, the body of the while loop will be executed.
- Once it is no longer true, the body will be bypassed.
- The first thing that happens is that the expression is checked, before the while loop is executed.

THIS ORDER IS IMPORTANT TO REMEMBER!

Using a While Loop

- The Syntax of the While Loop is:

```
while (loop repetition condition)
        <body>
```

- Where, the <body> is either one statement followed by a semicolon or a compound statement surrounded by {}.
- Remember the body is only executed when the condition is true.
- Then, after the body is executed, the condition is tested again...

Using a While Loop

- Notice, you must remember to initialize the loop control variable before you enter the while loop.
- Then, you must have some way of updating that variable inside of the body of the loop so that it can change the condition from true to false at some desired time.
- If this last step is missing, the loop will execute "forever" ... this is called an infinite loop.

Using a While Loop

- **We will need a <u>control variable</u> to be used to determine when the loop is done...**

```
char response = ‘n’;
while (‘n’ == response)
{
      cout << “Please enter ... “;
      cin >> data;                cin.get();
      cout << “We received: “ << data
                  << “\nIs this correct? (y/n)”;
      cin >> response;      cin.get();

}
```

Using a While Loop

- **What is a drawback of the previous loop?**
  - The user may have entered a lower or upper case response!
- One way to fix this:
  - Change the logical expression to list all of the legal responses

```
while ('n' == response || 'N' == response)
{
   ...
}
```

Using a While Loop

- **Yet another** way to fix this:
  - To loop, assuming that they want to continually try again until they enter a Y or a y!
  - Notice the use of AND versus OR!

```
while (‘y’ != response && ‘Y’ != response)
{
   ...
}
```

Using a While Loop

- **Another** way to fix this:
  - Use the tolower function in the ctype.h library:

```
#include <cctype>

while (tolower(response) != ‘y’)
{
   ...
}
```

Using a While Loop

- **Another** way to fix this:
  - Use the toupper function in the ctype.h library:

```
#include <cctype>

while (toupper(response) != ‘Y’)
{
   ...
}
```

Using a do while Loop

- This same loop could have been rewritten using a do while loop instead
- do while loops have the form: (notice semicolons!)

```
do
        single statement;
while (logical expression);
```

```
do
{
        many statements;
} while (logical expression);
```

Using a do while Loop

- Things to notice about a do while statement:

  (1) The body of a do while statement can be one statement or a compound statement surrounded by {}

  (2) Each statement in the do while loop is separated by a semicolon

  (3) Notice the body is always executed once! Even if the logical expression is false the first time!

Using a do while Loop

- **Don't use a do while unless you are sure that the body of the loop should be executed at least once!**

```
char response;
do {
     cout << “Please enter ... “;
     cin >> data;              cin.get();
     cout << “We received: “ << data
               << “\nIs this correct? (y/n)”;
     cin >> response;      cin.get();
} while (‘y’ != response && ‘Y’ != response);
```

Using a for loop

- **The for loop is commonly used to loop a certain number of times. For example, you can use it to print out the integers 1 thru 9:**

```
int i;

for (i=1; i <= 9; ++i)
    cout << i << endl;
```

Using a for loop

- i is called the loop control variable.
- It is most common to use variables i, j, and k for control variables.
- But, mnemonic names are better!

```
for (initialize; conditional exp; increment)
                <body>
```

- **Note: The body of the for loop is either one statement followed by a semicolon or a compound statement surrounded by {}.**

Using a for loop

- The for statement will first

  (1) INITIALIZE VARIABLE i to 1;

  (2) Check the logical expression to see if it is True or False;

  (3) if it is True the body of the loop is executed and it INCREMENTs VARIABLE i by 1;

  or, if it is False the loop is terminated and the statement following the body of the loop is executed.

Using a do while Loop

- When using loops, desk check for the following conditions:

  (1) Has the loop iterated one too many times? Or, one too few times?

  (2) Have you properly initialized the variables used in your while or do-while logical expressions?

  (3) Are you decrementing or incrementing those variables within the loop?

  (4) Is there an infinite loop?

Giving the user another chance

- **Remember our program from last class**
- **We displayed error messages if they entered in an unusual number – for the number of classes they were taking.**
- **This time, we will still display that error message, but we will loop until a valid number has been entered.**

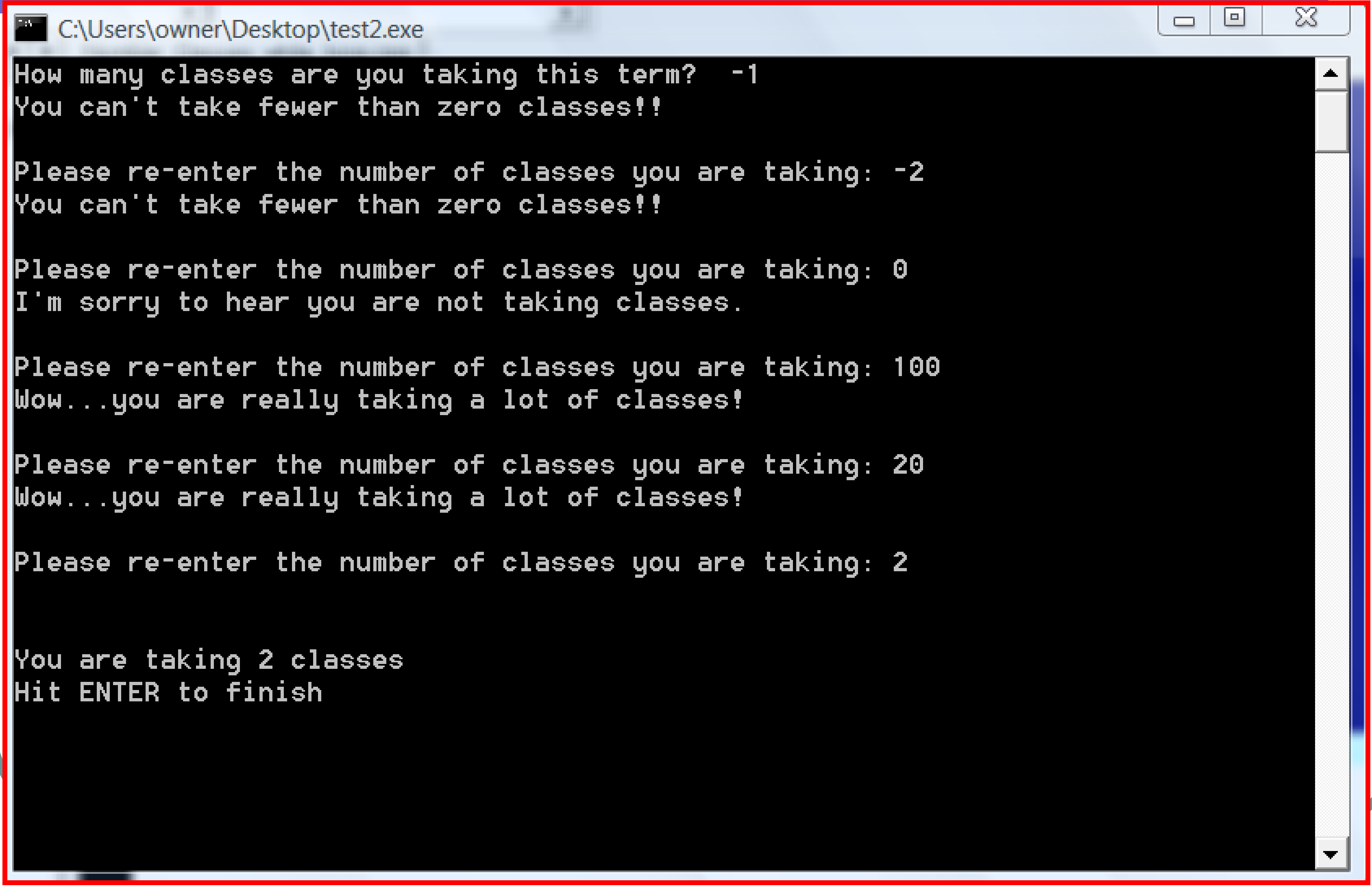
Giving the User another chance

```
//Written by: Karla Fant
//Purpose: To demonstrate the use of loops
//
//This program asks the user how many classes they
//are taking. It will continue to loop until a valid
//input value is received.

#include <iostream>
using namespace std;

int main()
{
    int num_classes = 0;  //the number of classes you are taking

    //prompt and read in the number of classes
    cout << "How many classes are you taking this term?  ";
    cin >> num_classes;
    cin.get();
```

Giving the User another chance

```
//check to see if the user entered a reasonable number
    while (0 >= num_classes || 5 < num_classes)     //out of range!
    {
       if (0 > num_classes)
          cout << "You can't take fewer than zero classes!!\n\n";
       else if (0 == num_classes)   //zero classes!
          cout << "I'm sorry to hear you are not taking classes.\n\n";
       else if (5 < num_classes)     //more than 5 classes
         cout << "Wow...you are really taking a lot of classes!\n\n";

       cout << "Please re-enter the number of classes you are taking: ";
       cin >> num_classes;
       cin.get();
    }

    //At this point, we KNOW that we have a valid number...
    cout << "\n\nYou are taking " << num_classes << " classes"
          <<endl;
    cout << "Hit ENTER to finish";
```

Looping until you get it right!...


C:\Users\owner\Desktop\test2.exe

```
How many classes are you taking this term?  -1
You can't take fewer than zero classes!!

Please re-enter the number of classes you are taking: -2
You can't take fewer than zero classes!!

Please re-enter the number of classes you are taking: 0
I'm sorry to hear you are not taking classes.

Please re-enter the number of classes you are taking: 100
Wow...you are really taking a lot of classes!

Please re-enter the number of classes you are taking: 20
Wow...you are really taking a lot of classes!

Please re-enter the number of classes you are taking: 2


You are taking 2 classes
Hit ENTER to finish
```

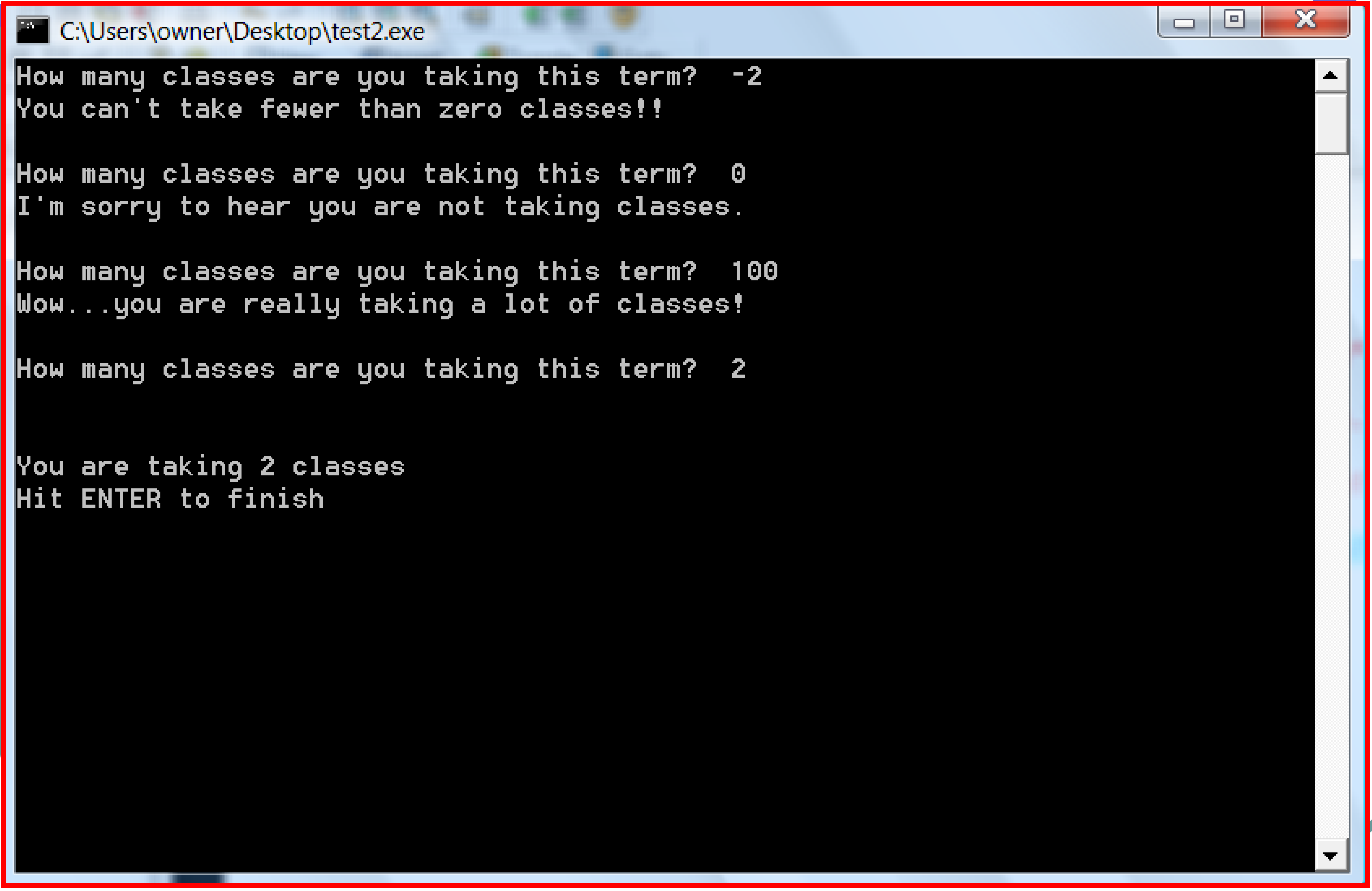
Using the do-while loop instead

- A do-while loop means that we will always use the loop once, but we don't know if we will use it more than once:
- Pseudo code…

```
do
{
    // read in the value
    // display error messages if a bad value is received
}while (the value is still bad);
```

```
//prompt and read in the number of classes
do
{
      cout << "How many classes are you taking this term?  ";
      cin >> num_classes;
      cin.get();

      //check to see if the user entered a reasonable number
      if (0 > num_classes)
        cout << "You can't take fewer than zero classes!!\n\n";
      else if (0 == num_classes)   //zero classes!
        cout << "I'm sorry to hear you are not taking classes.\n\n";
      else if (5 < num_classes)    //more than 5 classes
        cout << "Wow...you are really taking a lot of classes!\n\n";

} while (0 >= num_classes || 5 < num_classes);     //out of range!

//At this point, we KNOW that we have a valid number...

//echo what we got back to the user
cout << endl <<endl;
cout << "You are taking " << num_classes << " classes"
    <<endl;
```

Running this new version…



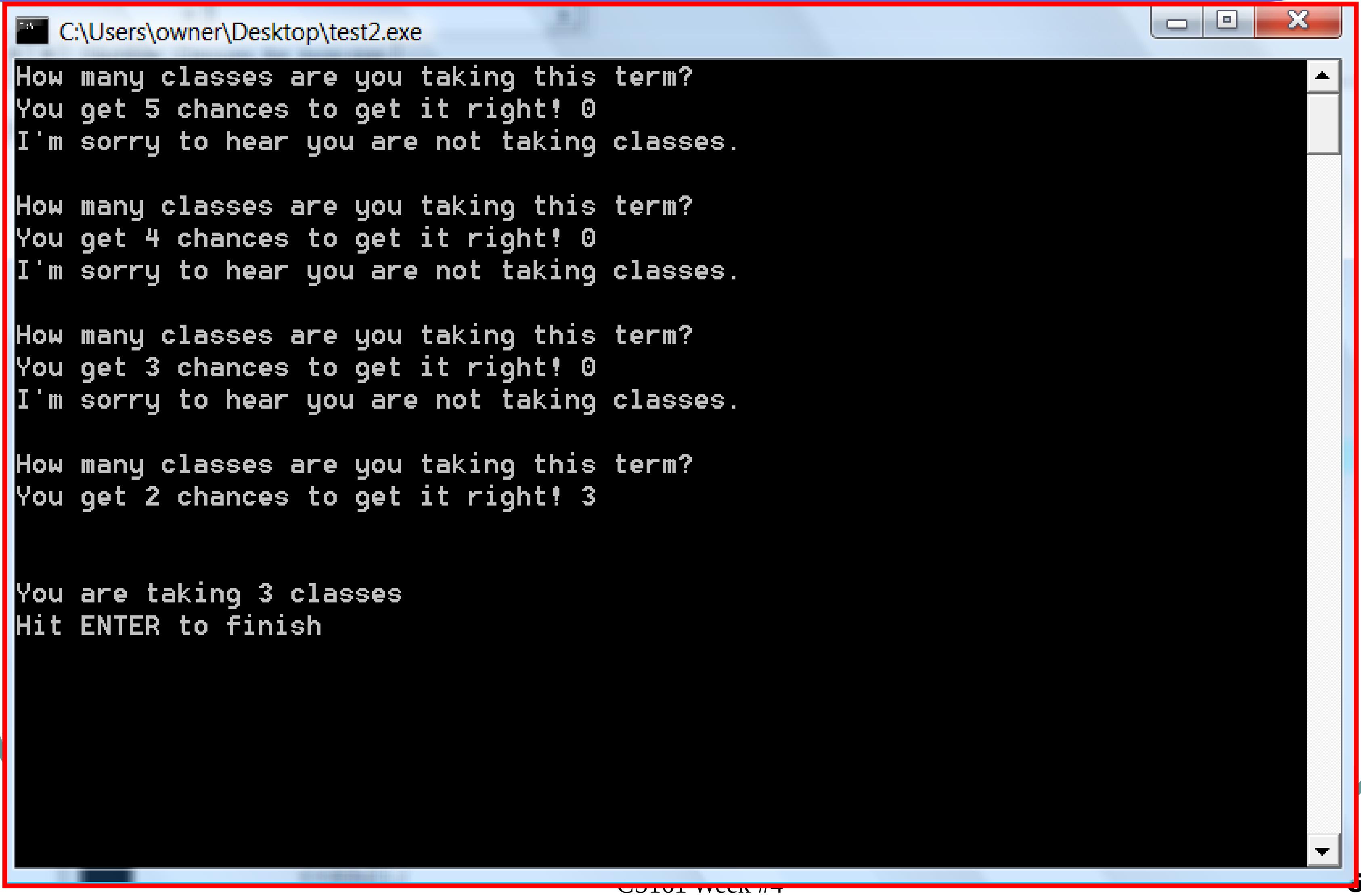
C:\Users\owner\Desktop\test2.exe

```
How many classes are you taking this term?  -2
You can't take fewer than zero classes!!

How many classes are you taking this term?  0
I'm sorry to hear you are not taking classes.

How many classes are you taking this term?  100
Wow...you are really taking a lot of classes!

How many classes are you taking this term?  2


You are taking 2 classes
Hit ENTER to finish
```

Give them only 5 chance….

```
 bool out_of_range = true;

for (int i = 5; i > 0 && out_of_range; --i)
{
    //prompt and read in the number of classes
    cout << "How many classes are you taking this term?  ";
    cout << "\nYou get " <<i <<" chances to get it right! ";
    cin >> num_classes;
    cin.get();

    //check to see if the user entered a reasonable number
    if (0 < num_classes && 5 >= num_classes)
       out_of_range = false;
    if (0 > num_classes)
       cout << "You can't take fewer than zero classes!!\n\n";
    else if (0 == num_classes)    //zero classes!
       cout << "I'm sorry to hear you are not taking classes.\n\n";
    else if (5 < num_classes)     //more than 5 classes
      cout << "Wow...you are really taking a lot of classes!\n\n";
```

Running this new version…

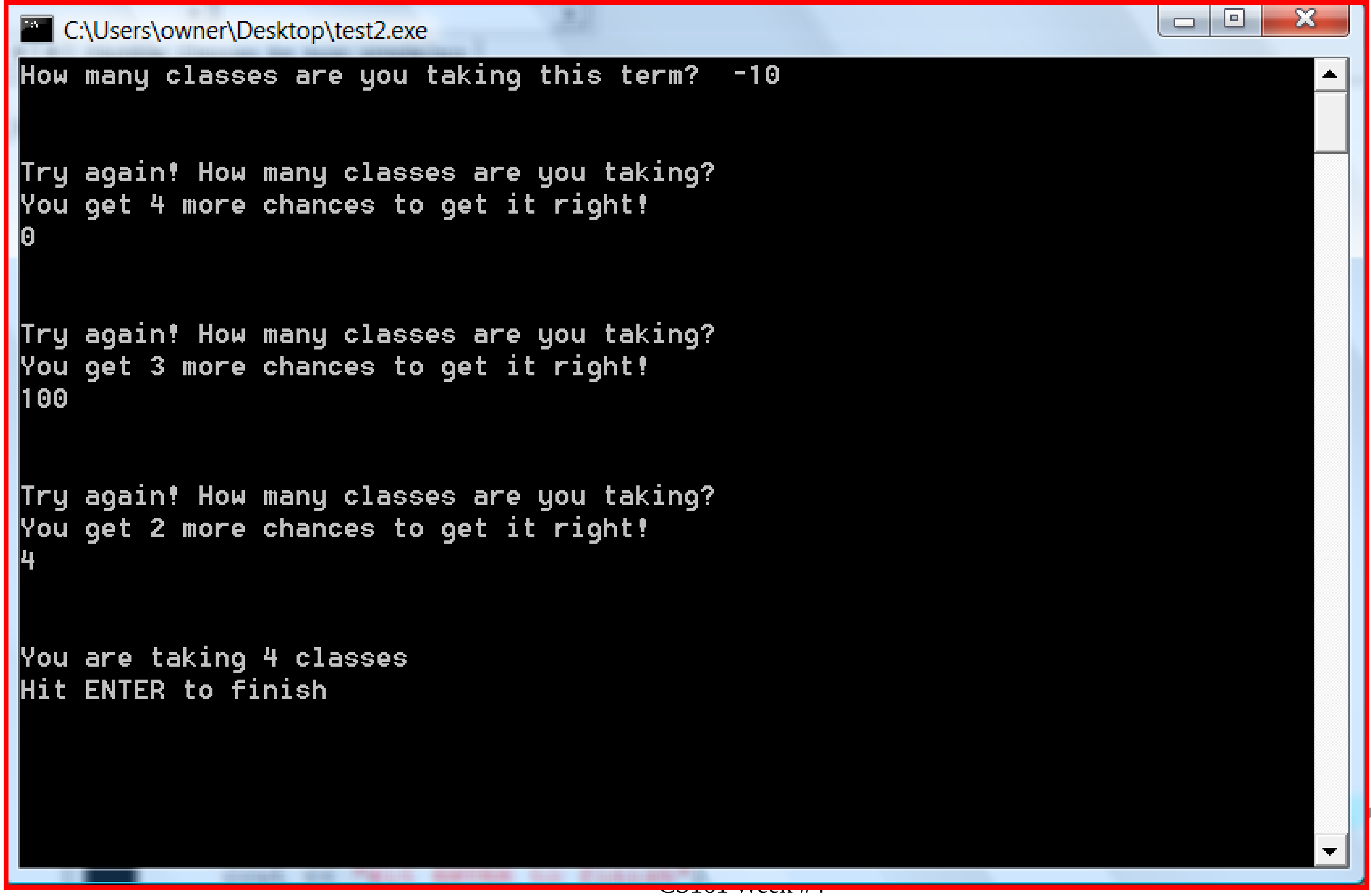


```
How many classes are you taking this term?
You get 5 chances to get it right! 0
I'm sorry to hear you are not taking classes.

How many classes are you taking this term?
You get 4 chances to get it right! 0
I'm sorry to hear you are not taking classes.

How many classes are you taking this term?
You get 3 chances to get it right! 0
I'm sorry to hear you are not taking classes.

How many classes are you taking this term?
You get 2 chances to get it right! 3


You are taking 3 classes
Hit ENTER to finish
```

Breaking it down….simplier

```
//prompt and read in the number of classes
    cout << "How many classes are you taking this term?  ";
    cin >> num_classes;
    cin.get();

    for (int i = 1; i < 5 && (num_classes <=0 || num_classes > 5); ++i)
    {
        cout <<"\n\nTry again! How many classes are you taking? " <<endl;
        cout <<"You get " <<5-i <<" more chances to get it right!\n";
        cin >> num_classes;
        cin.get();
    }
```

Running this new version…



```
How many classes are you taking this term?  -10


Try again! How many classes are you taking?
You get 4 more chances to get it right!
0


Try again! How many classes are you taking?
You get 3 more chances to get it right!
100


Try again! How many classes are you taking?
You get 2 more chances to get it right!
4


You are taking 4 classes
Hit ENTER to finish
```

Using Graphics…movement

To achieve movement, we need to draw and un-draw our objects over and over at different locations

```
//Here is where I am going to put my variables
   int window_size;
   int color;
   int circle_radius;
   int num_circles;      //number of circles to draw & redraw
   int background_color;  //how to undraw a circle!
   int slow;             //allow user to set a delay

   cout << "How big of a window do you want (pick a number less than 1200): ";
   cin >> window_size;   cin.get();
   initwindow(window_size, window_size);
```

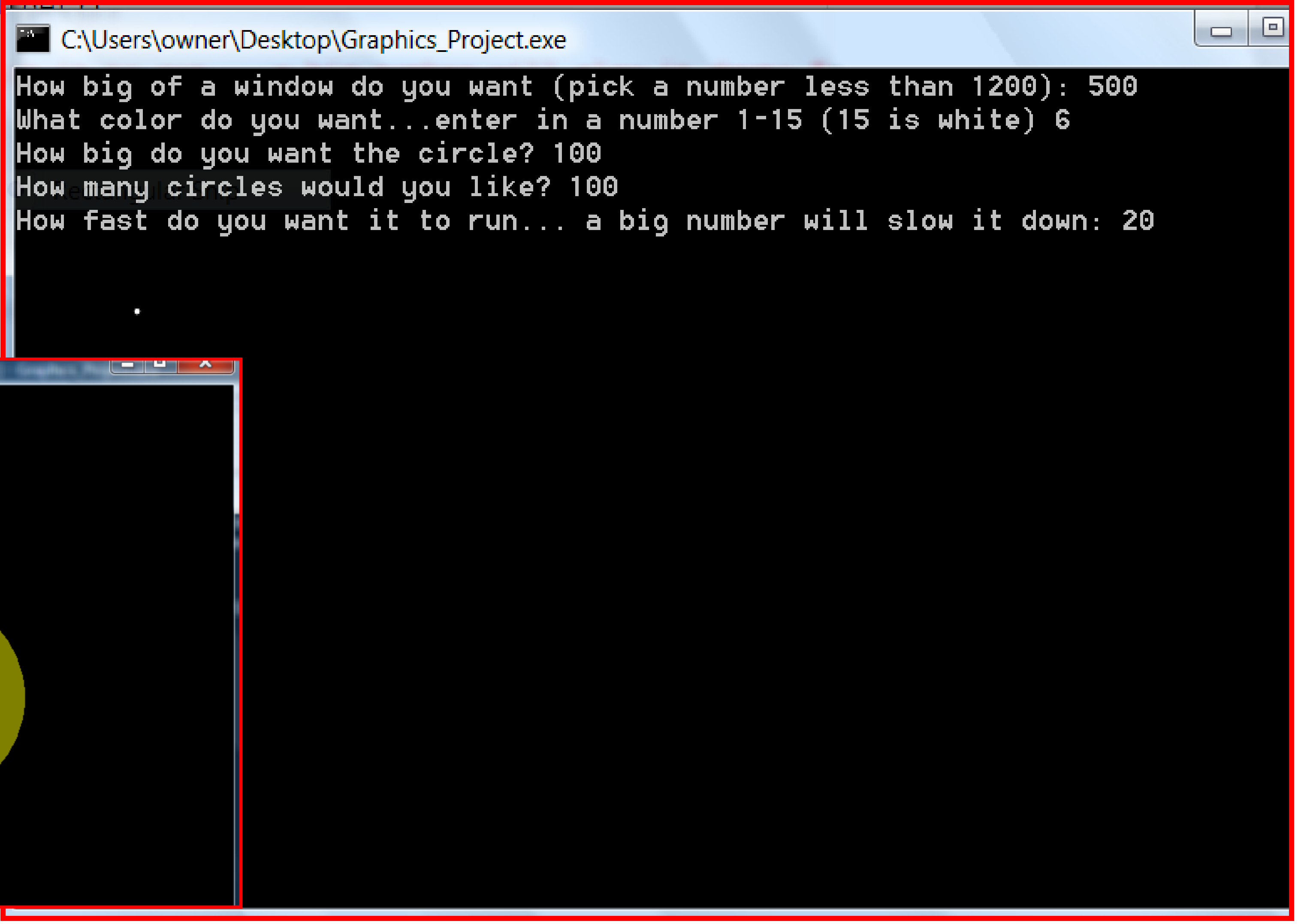
Using Graphics…movement

```
//set up the drawing colors for outlines and fills
cout << "What color do you want...enter in a number 1-15 (15 is white) ";
cin >> color;  cin.get();
setcolor(color);
setfillstyle(SOLID_FILL,color);
background_color = getbkcolor( ); //get the background color

cout << "How big do you want the circle? ";
cin >> circle_radius;   cin.get();

cout << "How many circles would you like? ";
cin >> num_circles;     cin.get();

cout << "How fast do you want it to run... a big number will slow it down: ";
cin >> slow;                  cin.get();
```

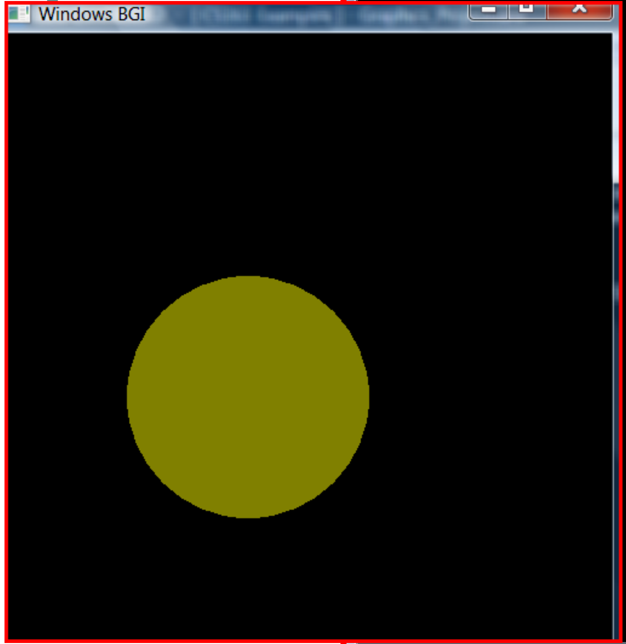
Using Graphics…movement

```
//Draw the first ellipse
int x = 0;          //let's start at the lower left corner of the window
int y = window_size;
fillellipse(x,y,circle_radius,circle_radius);

for (int i = 0; i < num_circles; ++i)
{
   delay(slow);  //if you don't do this…it runs too fast!
   //undraw the circle
   setcolor(background_color);  //really make it disappear!
   setfillstyle(SOLID_FILL,background_color);
   fillellipse(x,y,circle_radius,circle_radius);  //undraw the circle

   x = x + 5; //same as x += 5;      //go up diagonally
   y = y - 5; //same as y -=  5;

   setcolor(color);  //go back to requested color
   setfillstyle(SOLID_FILL,color);
   fillellipse(x,y,circle_radius,circle_radius);
}
```

Running this new version…



```
How big of a window do you want (pick a number less than 1200): 500
What color do you want...enter in a number 1-15 (15 is white) 6
How big do you want the circle? 100
How many circles would you like? 100
How fast do you want it to run... a big number will slow it down: 20
```



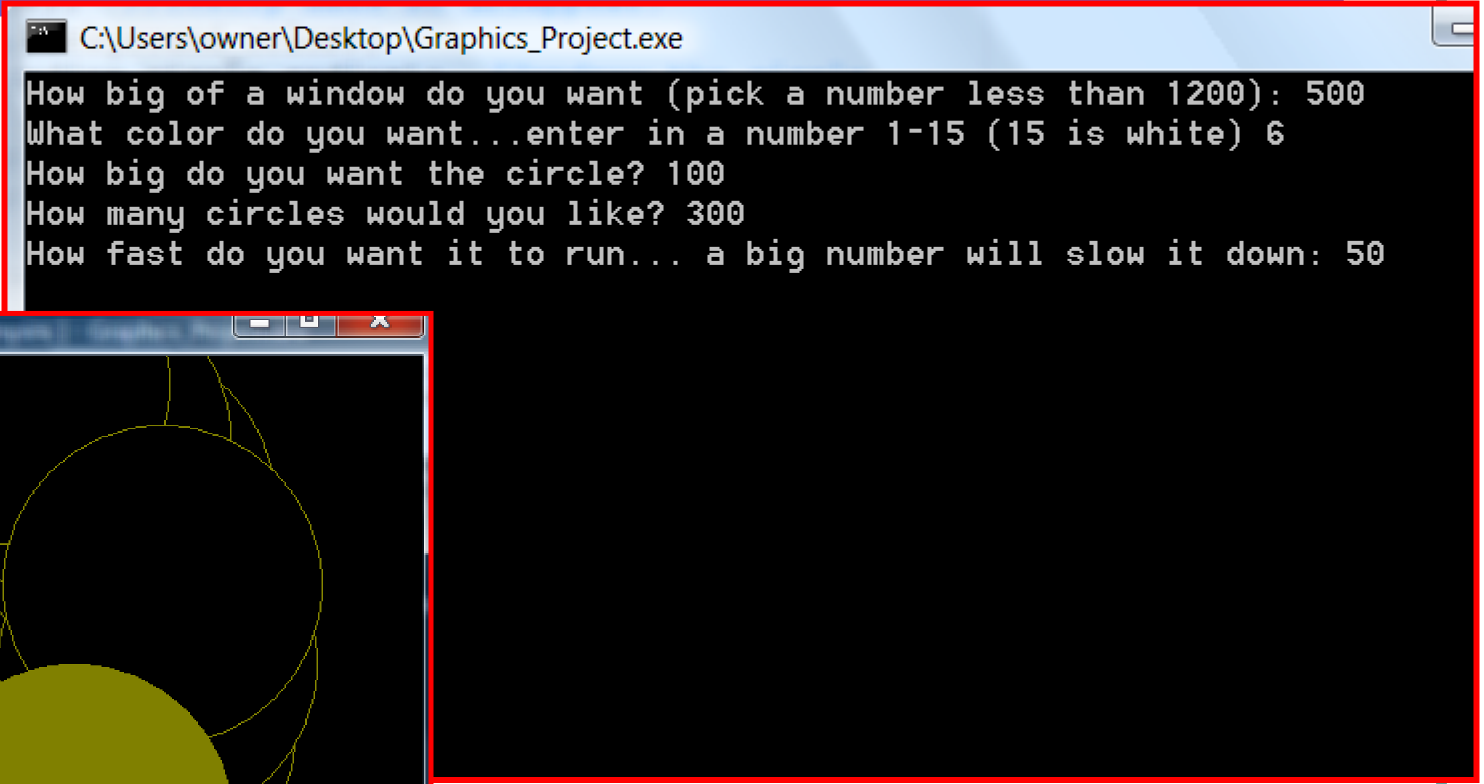
Moving in a spiral like motion…

```
for (int i = 0; i < num_circles; ++i)
   {
      delay(slow);
     // setcolor(background_color);  //make outline disappear
      setfillstyle(SOLID_FILL,background_color);
      fillellipse(x,y,circle_radius,circle_radius);  //undraw the circle

      x += angle * cos(angle);
      y += angle *sin(angle);
      angle += 1;
      if (angle > 360) angle = 0;


           //continued on next page….
```

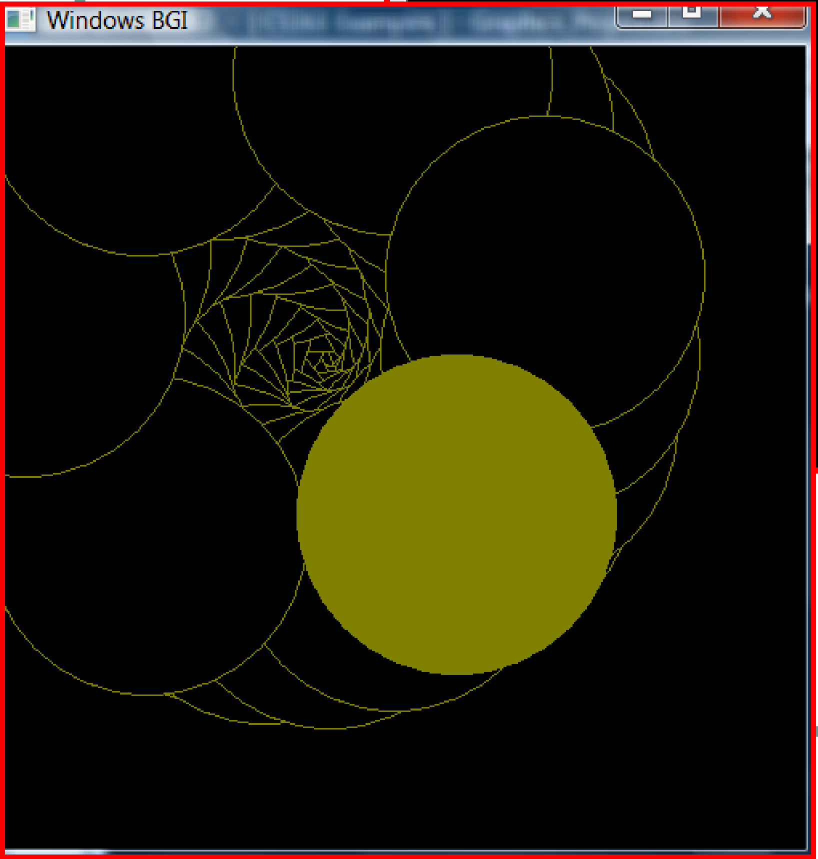
Moving in a spiral like motion…

```
//Reset if they go outside of the window back to the center
        if ((x > window_size || x <0) || (y > window_size || y < 0))
        {
           x = window_size/2;
           y = window_size/2;
           angle = 0;
        }

        setcolor(color);  //go back to requested color
        setfillstyle(SOLID_FILL,color);
        fillellipse(x,y,circle_radius,circle_radius);
    }
```

Running this new version…


C:\Users\owner\Desktop\Graphics_Project.exe

```
How big of a window do you want (pick a number less than 1200): 500
What color do you want...enter in a number 1-15 (15 is white) 6
How big do you want the circle? 100
How many circles would you like? 300
How fast do you want it to run... a big number will slow it down: 50
```


Windows BGI