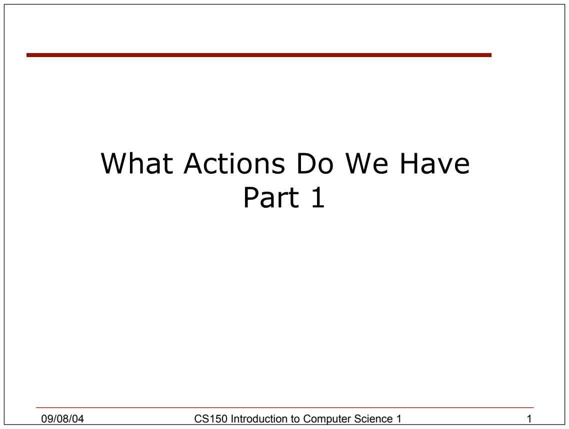
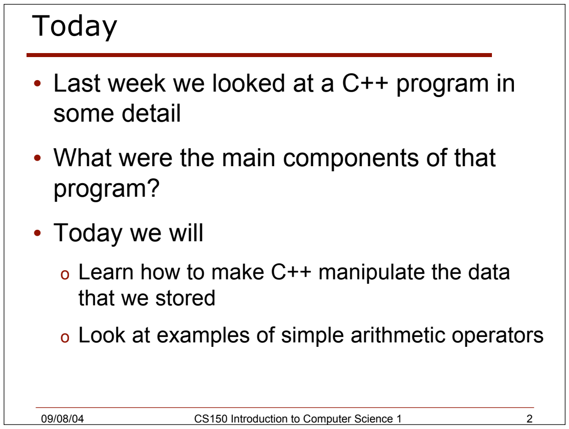
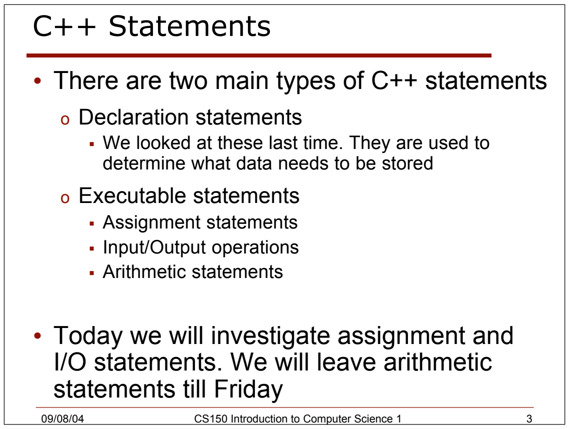


What Actions Do We Have
Part 1



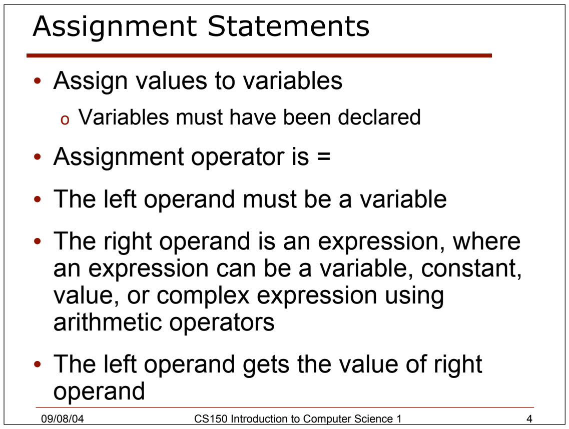
Today

- Last week we looked at a C++ program in some detail
- What were the main components of that program?
- Today we will
  - Learn how to make C++ manipulate the data that we stored
  - Look at examples of simple arithmetic operators



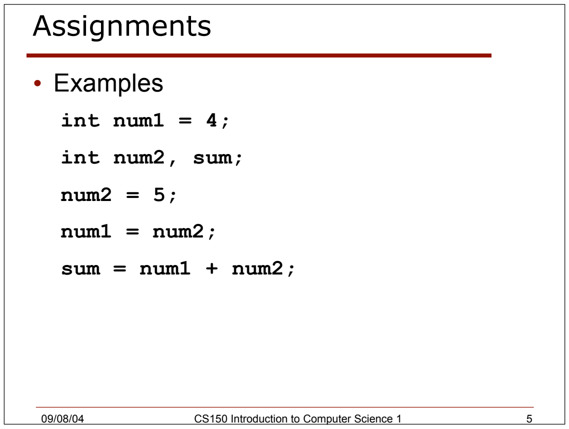
C++ Statements

- There are two main types of C++ statements
  - Declaration statements
    - We looked at these last time. They are used to determine what data needs to be stored
  - Executable statements
    - Assignment statements
    - Input/Output operations
    - Arithmetic statements
- Today we will investigate assignment and I/O statements. We will leave arithmetic statements till Friday



Assignment Statements

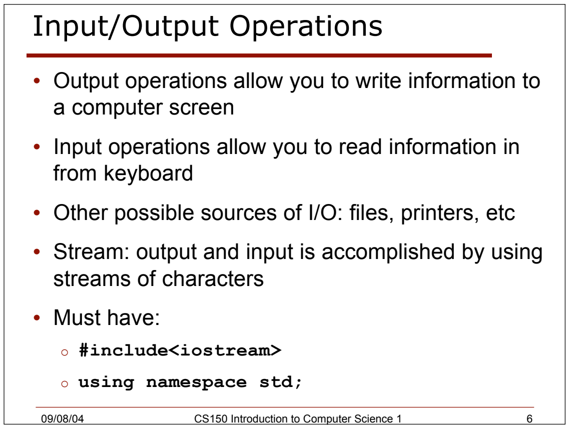
- Assign values to variables
  - Variables must have been declared
- Assignment operator is =
- The left operand must be a variable
- The right operand is an expression, where an expression can be a variable, constant, value, or complex expression using arithmetic operators
- The left operand gets the value of right operand



Assignments

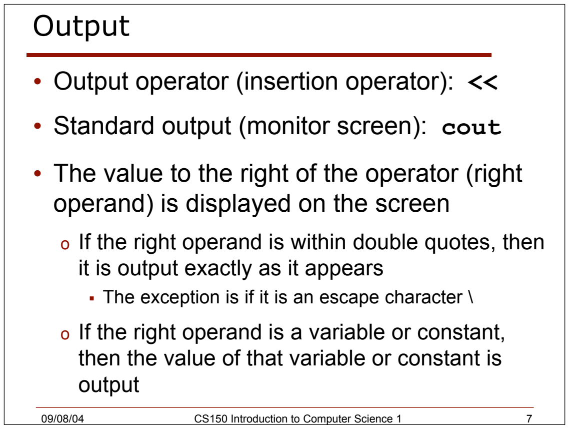
- Examples

```
int num1 = 4;
int num2, sum;
num2 = 5;
num1 = num2;
sum = num1 + num2;
```



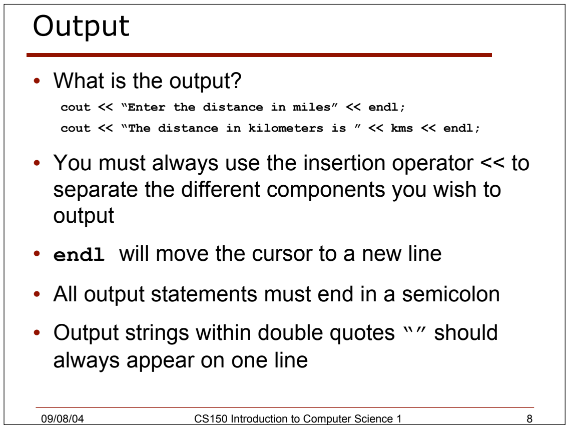
Input/Output Operations

- Output operations allow you to write information to a computer screen
- Input operations allow you to read information in from keyboard
- Other possible sources of I/O: files, printers, etc
- Stream: output and input is accomplished by using streams of characters
- Must have:
  - `#include<iostream>`
  - `using namespace std;`



Output

- Output operator (insertion operator): **<<**
- Standard output (monitor screen): **`cout`**
- The value to the right of the operator (right operand) is displayed on the screen
  - If the right operand is within double quotes, then it is output exactly as it appears
    - The exception is if it is an escape character \
  - If the right operand is a variable or constant, then the value of that variable or constant is output

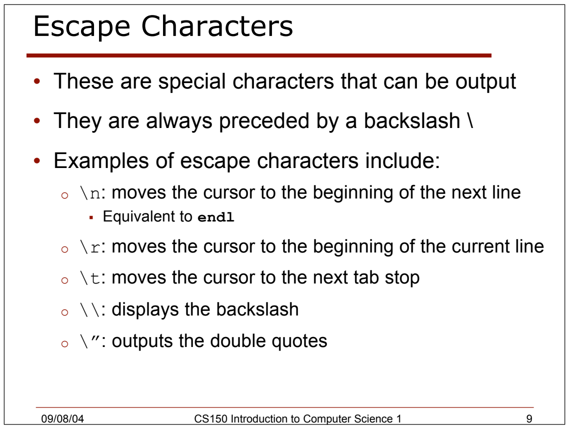


Output

- What is the output?

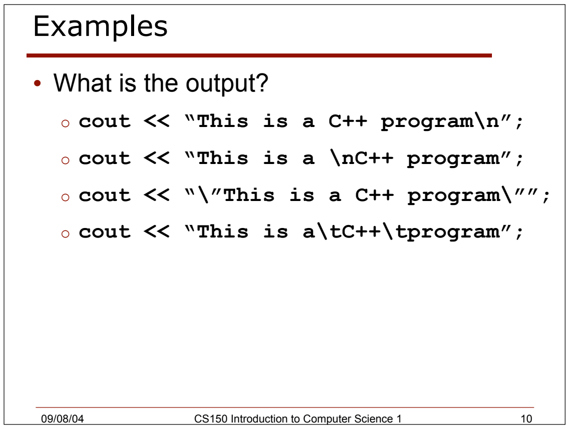
```
cout << "Enter the distance in miles" << endl;
cout << "The distance in kilometers is " << kms << endl;
```

- You must always use the insertion operator << to separate the different components you wish to output
- **`endl`** will move the cursor to a new line
- All output statements must end in a semicolon
- Output strings within double quotes "" should always appear on one line



Escape Characters

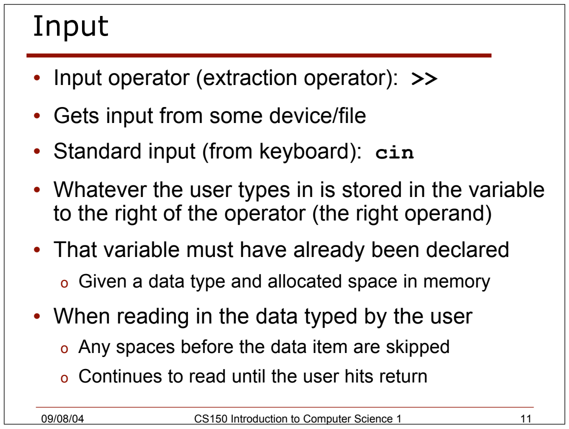
- These are special characters that can be output
- They are always preceded by a backslash \
- Examples of escape characters include:
  - `\n`: moves the cursor to the beginning of the next line
    - Equivalent to **`endl`**
  - `\r`: moves the cursor to the beginning of the current line
  - `\t`: moves the cursor to the next tab stop
  - `\\`: displays the backslash
  - `\"`: outputs the double quotes



Examples

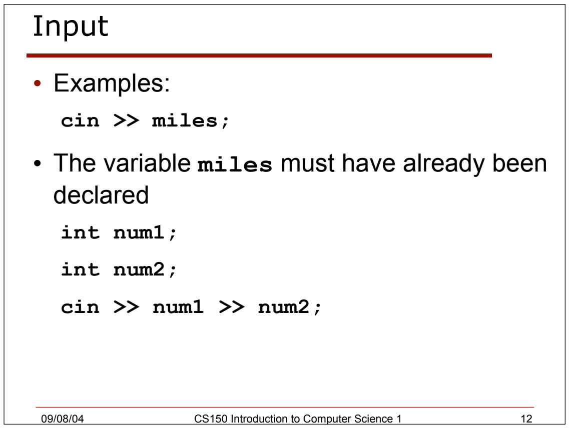
- What is the output?

```
cout << "This is a C++ program\n";
cout << "This is a \nC++ program";
cout << "\"This is a C++ program\"";
cout << "This is a\tC++\tprogram";
```



Input

- Input operator (extraction operator): **>>**
- Gets input from some device/file
- Standard input (from keyboard): **`cin`**
- Whatever the user types in is stored in the variable to the right of the operator (the right operand)
- That variable must have already been declared
  - Given a data type and allocated space in memory
- When reading in the data typed by the user
  - Any spaces before the data item are skipped
  - Continues to read until the user hits return



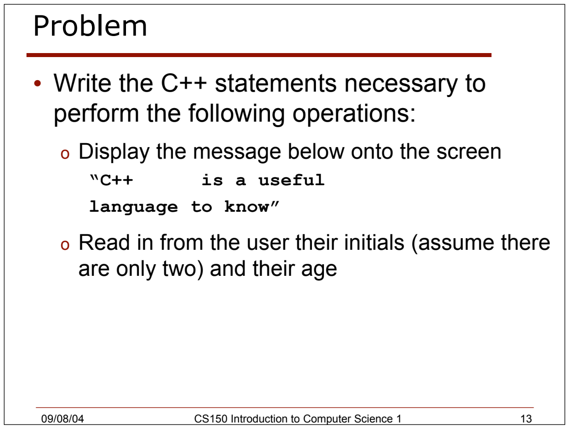
Input

- Examples:

```
cin >> miles;
```

- The variable **`miles`** must have already been declared

```
int num1;
int num2;
cin >> num1 >> num2;
```

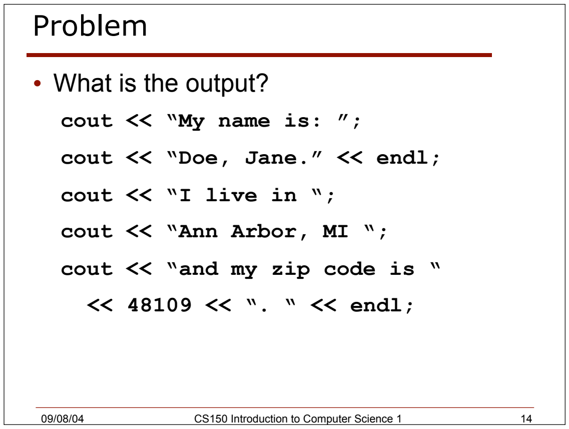


Problem

- Write the C++ statements necessary to perform the following operations:
  - Display the message below onto the screen

```
“C++      is a useful
language to know”
```

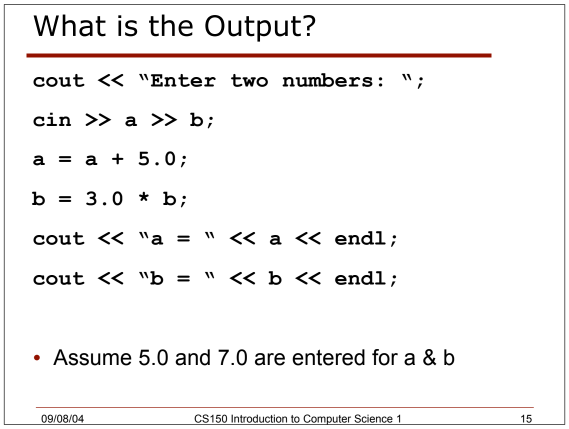
  - Read in from the user their initials (assume there are only two) and their age



Problem

- What is the output?

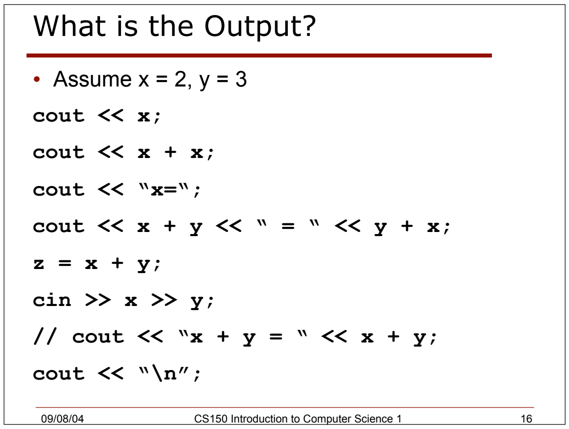
```
cout << “My name is: ”;
cout << “Doe, Jane.” << endl;
cout << “I live in “;
cout << “Ann Arbor, MI “;
cout << “and my zip code is “
  << 48109 << “. “ << endl;
```



What is the Output?

```
cout << “Enter two numbers: “;
cin >> a >> b;
a = a + 5.0;
b = 3.0 * b;
cout << “a = “ << a << endl;
cout << “b = “ << b << endl;
```

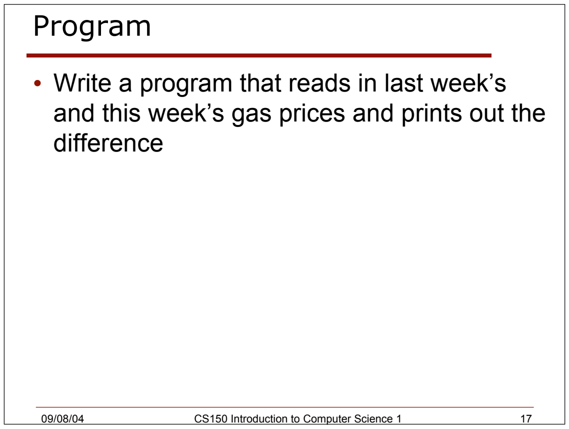
- Assume 5.0 and 7.0 are entered for a & b



What is the Output?

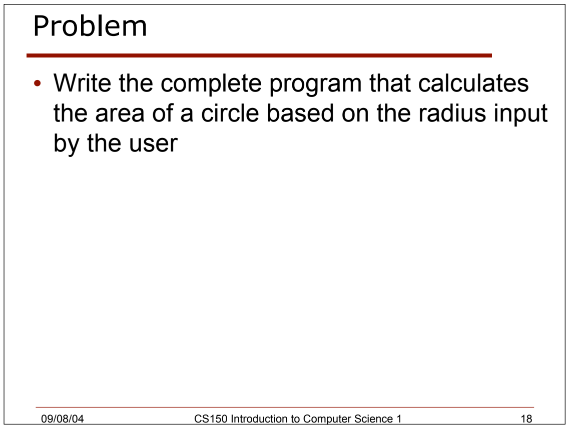
- Assume x = 2, y = 3

```
cout << x;
cout << x + x;
cout << “x=“;
cout << x + y << “ = “ << y + x;
z = x + y;
cin >> x >> y;
// cout << “x + y = “ << x + y;
cout << “\n”;
```



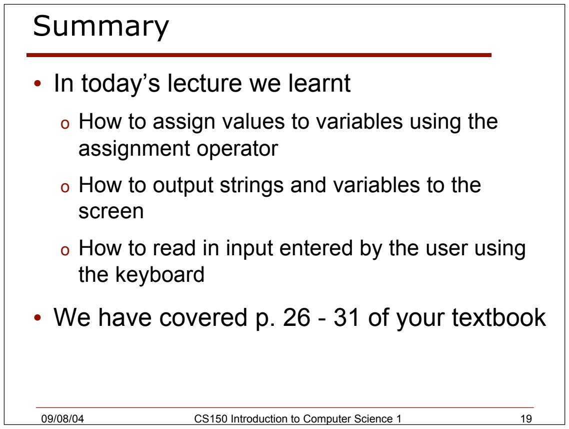
Program

- Write a program that reads in last week’s and this week’s gas prices and prints out the difference



Problem

- Write the complete program that calculates the area of a circle based on the radius input by the user



Summary

- In today's lecture we learnt
  - How to assign values to variables using the assignment operator
  - How to output strings and variables to the screen
  - How to read in input entered by the user using the keyboard
- We have covered p. 26 - 31 of your textbook