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
  - o Learn how to make C++ manipulate the data that we stored
  - o Look at examples of simple arithmetic operators

Example

- Can you spot what is incorrect in the following program:

```
int main()
{
  const int pi = 3.14;
  double num;
  int i,j;

  num = e2;
  i = 4,000;
  ch = "b"; j = i;
  pi = 5;

  return 0;
}
```

Problem

- Write a program that asks the user to enter the radius of a circle and then computes and displays the circle's area
- What constant declarations does our program need?
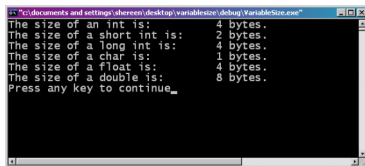- What variable declarations does out program need?

Variable Sizes

- On my machine the sizes are

Variable Size Program

```
#include "stdafx.h"
#include <iostream>

using namespace std;

int main()
{
    cout << "The size of an int is:\t\t"    << sizeof(int)    << "
bytes.\n";
    cout << "The size of a short int is:\t" << sizeof(short)  << "
bytes.\n";
    cout << "The size of a long int is:\t"  << sizeof(long)   << "
bytes.\n";
    cout << "The size of a char is:\t\t"    << sizeof(char)   << "
bytes.\n";
    cout << "The size of a float is:\t\t"   << sizeof(float)  << "
bytes.\n";
    cout << "The size of a double is:\t"    << sizeof(double) << "
bytes.\n";

    return 0;
}
```

Variable Ranges

Type	Size	Values
int	4 bytes	−2,147,483,648 to 2,147,483,647
short int	2 bytes	−32,768 to 32,767
long int	4 bytes	0 to 4,294,967,295
char	1 byte	256 character values
float	4 bytes	1.2e–38 to 3.4e38
double	8 bytes	2.2e–308 to 1.8e308

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