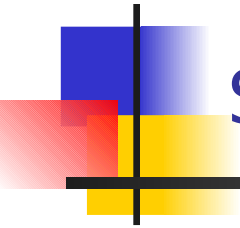
CSCI 152
Programming Fundamentals II

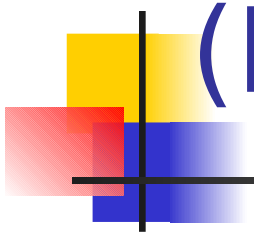


Summer 2008

Basics C++ Elements &
Basic Input / Output

June 5, 2008

Objectives (Basic Elements Ch #2)



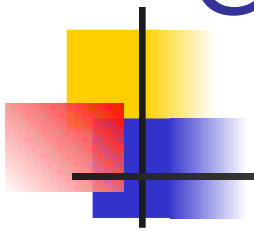
In this lecture/chapter you should review and:

- Become familiar with the basic components of a C++ program, including functions, special symbols, and identifiers
- Explore simple data types and examine the string data type
- Discover how to use arithmetic operators
- Examine how a program evaluates arithmetic expressions
- Learn what an assignment statement is and what it does

Introduction

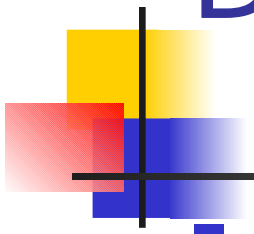
- Computer program: sequence of statements designed to accomplish some task
- Programming: planning/creating a program
- Syntax: rules that specify which statements (instructions) are legal
- Programming language: a set of rules, symbols, and special words

C++ Programs



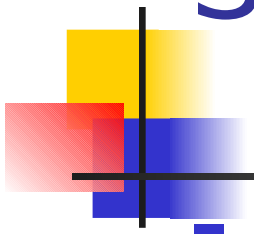
- A C++ program is a collection of one or more subprograms, called functions
- A subprogram or a function is a collection of statements that, when activated (executed), accomplishes something
- Every C++ program has a function called main
- The smallest individual unit of a program written in any language is called a token

Data Types



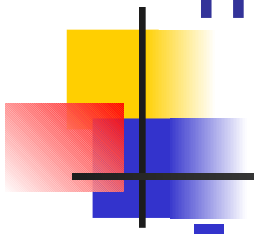
- Data Type: set of values together with a set of operations is called a data type
- C++ data can be classified into three categories:
  - Simple data type
  - Structured data type
  - Pointers

Simple Data Types



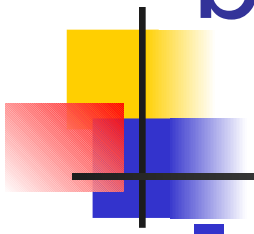
- Three categories of simple data
  - Integral: integers (numbers without a decimal)
  - Floating-point: decimal numbers
  - Enumeration type: user-defined data type

int Data Type



- Examples:
  - -6728
  - 0
  - 78
- Positive integers do not have to have a + sign in front of them
- No commas are used within an integer
- Commas are used for separating items in a list

bool Data Type

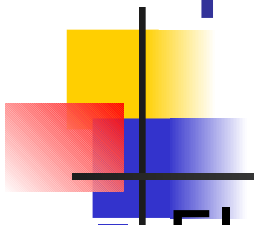


- bool type
  - Has two values, true and false
  - Manipulate logical (Boolean) expressions
- True and false are called logical values
- bool, true, and false are reserved

char Data Type

- The smallest integral data type
- Used for characters: letters, digits, and special symbols
- Each character is enclosed in single quotes
- Some of the values belonging to char data type are: 'A', 'a', '0', '*', '+', '$', '&'
- A blank space is a character and is

Floating-Point Data Types

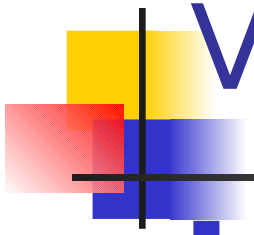


- Float: represents any real number
  - Range: -3.4E+38 to 3.4E+38
- Memory allocated for the float type is 4 bytes
- Double: represents any real number
  - Range: -1.7E+308 to 1.7E+308
- Memory allocated for double type is 8 bytes

string Data Type

- Programmer-defined type supplied in standard library
- Sequence of zero or more characters
- Enclosed in double quotation marks
- Null: a string with no characters
- Each character has relative position in string
- Position of first character is 0, the position of the second is 1, and so on
- Length: number of characters in string

Declaring & Initializing Variables

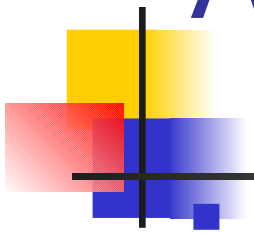


- Variables can be initialized when declared:

```
int first=13, second=10;

char ch=' ';

double x=12.6, y=123.456;
```

- first and second are integers with the values 13 and 10 respectively
- ch is a char whose value is empty
- x and y are doubles with 12.6 and 123.456

Assignment Statement



- The assignment statement takes the form:

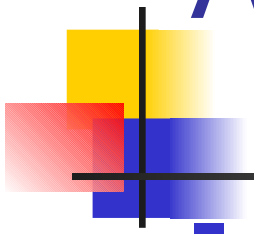
  variable = expression;

- Expression is evaluated and its value is assigned to the variable on the left side
- In C++ = is called the assignment operator
- A C++ statement such as:

  i = i + 2;

  evaluates whatever is in i, adds two to it, and assigns the new value to the

Arithmetic Operators

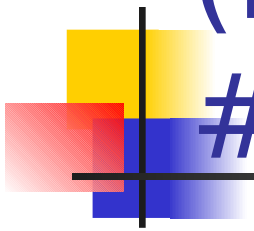


- C++ Operators
  - + addition
  - - subtraction
  - * multiplication
  - / division
  - % remainder (mod operator)
- +, -, *, and / can use with integral and floating-point data types
- Unary operator - has only one operand
- Binary Operator - has two

Example: Declaring and Assig of Simple data types

```
// Author:  Joe Student
// Program: Example Simple data Types
// Class:   CSci 152, Summer 2008
// Date:    June 5, 2008
//
// Desc:    Example of declaring and assigning the simple C++ data types
#include <iostream>
#include <string>

using namespace std;

int main(int argc, char** argv)
{
  int x, y=32;

  bool flag=true;

  char c='A';

  const float PI = 3.1415926;
  float radius = 1.5;
  double area;

  string name;

  area = PI * radius * radius;
```

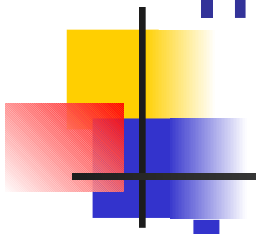
Objectives (Basic Input/Output Ch #2/#3)



In this lecture/chapter you should review and:

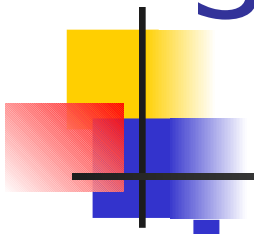
- Discover how to input data into memory using input statements
- Examine ways to output results using output statements
- Learn what a stream is and examine input and output streams
- Explore how to read data from the standard input device
- Learn how to write data to the standard output device

Input/Output Streams



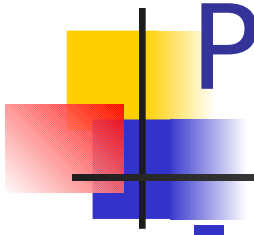
- I/O: sequence of bytes (stream of bytes) from source to destination
- Bytes are usually characters, unless program requires other types of information
- Stream: sequence of characters from source to destination
- Input Stream: sequence of characters from an input device to the computer
- Output Stream: sequence of characters from the computer to an output device

Standard I/O Devices



- Use iostream to extract (receive) data from keyboard and send output to the screen
- iostream contains definitions of two types
  - istream - input stream
  - ostream - output stream
- iostream has two variables
  - cin - stands for common input
  - cout - stands for common output

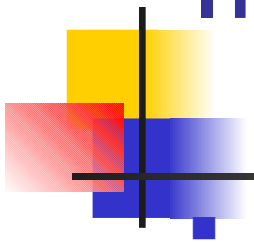
Using cin and cout in a Program and namespace



- cin and cout are declared in the header file iostream, but within a namespace named std
- To use cin and cout in a program, use the following two statements:

```
#include <iostream>

using namespace std;
```

Input (Read) Statement



- cin is used with >> to gather input

  cin>>variable>>variable. . .;

- The extraction operator is >>
- For example, if miles is a double variable

  cin>>miles;

  - Causes computer to get a value of type double
  - Places it in the memory cell miles

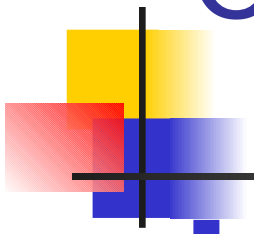
Input Statement (continued)

- Using more than one variable in cin allows more than one value to be read at a time
- For example, if feet and inch are variables of the type int a statement such as:

```
cin>>feet>>inch;
```

  - Inputs two integers from the keyboard
  - Places them in locations feet and inch respectively

Output



- The syntax of cout and << is:

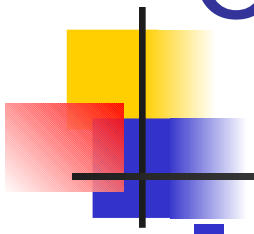
  cout<< expression or manipulator

  << expression or manipulator

  << ...;

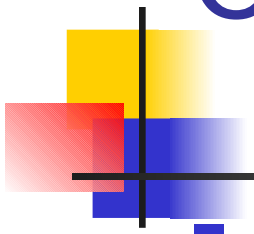
- Called an output (cout) statement
- The << operator is called the insertion operator or the stream insertion operator
- Expression evaluated and its value is printed at the current cursor position on

Output (continued)



- Manipulator: alters output
- endl: the simplest manipulator
  - Causes cursor to move to beginning of the next line

Output Example

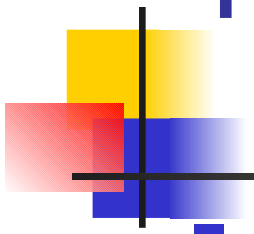


- Output of the C++ statement cout<<a; is meaningful if a has a value
  - For example, the sequence of C++ statements,

```
a = 45;

cout<<a;
```

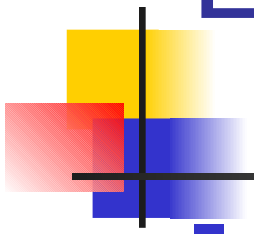
produces an output of 45

The New Line Character



- The new line character is ‘\n’
- Without this character the output is printed on one line
- Tells the output to go to the next line
- When \n is encountered in a string
  - Cursor is positioned at the beginning of next line
- A \n may appear anywhere in the

Examples



- Without the new line character:

```
cout<<"Hello there.";
cout<<"My name is Goofy.";
```

  - Would output:

```
Hello there.My name is Goofy.
```

- With the new line character:

```
cout<<"Hello there.\n";
cout<<"My name is Goofy.";
```

  - Would output

```
Hello there.
My name is Goofy.
```

Example of Basic I/O

```
// Author:  Joe Student
// Program: Example Simple data Types
// Class:   CSci 152, Summer 2008
// Date:    June 5, 2008
//
// Desc:    Example of declaring and assigning the simple C++ data types.
#include <iostream>
#include <string>

using namespace std;

int main(int argc, char** argv)
{
  int x;
  const float PI = 3.1415926;
  float radius = 1.5;
  double area;
  string name;

  cout << "What is your name: ";
  cin >> name;

  cout << "Enter an integer, and a floating point value: ";
  cin >> x >> radius;

  area = PI * radius * radius;

  cout << "Hello " << name << " you entered integer " << x
    << " and float " << radius << endl;

  cout << "The area of a circle using your float as the radius is "
    << area << endl;

}
```

Example of Basic I/O

```
What is your name: Derek
Enter an integer, and a floating point value: 42 3.55
Hello Derek you entered integer 42 and float 3.55
The area of a circle using your float as the radius is 39.5919
```