Computer Programing (CP)

Lecture # 4

Topics

- Introduction to C++ IDE
- Introduction to C++
- C++ programs
  - First
  - Named Constant
  - cin and cout using Strings

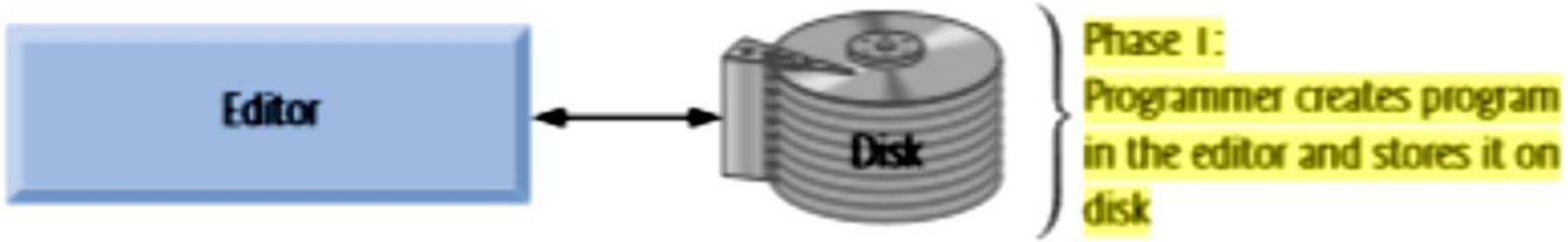
C++ Integrated Development Environment (IDE)

- C++ systems consist of three parts:
  - a program development environment,
  - the language and
  - the C++ Standard Library

- C++ programs typically go through six phases:
  - edit,
  - preprocess,
  - compile,
  - link,
  - load and
  - execute

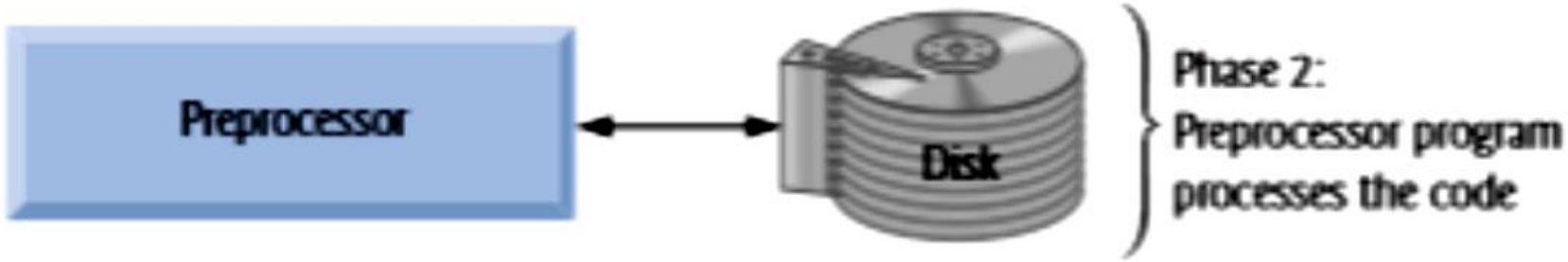
Phase 1: Creating a Program

- C++ program(source code) can be written using an editor
- Make any necessary corrections and save the program on a secondary storage device, such as your hard drive.
- C++ source code file names often end with the .cpp, .cxx, .cc or .C extensions(note that C is in uppercase)
- Popular IDEs include Microsoft® Visual Studio 2010 Express Edition,Dev C++, NetBeans, Eclipse and CodeLite.



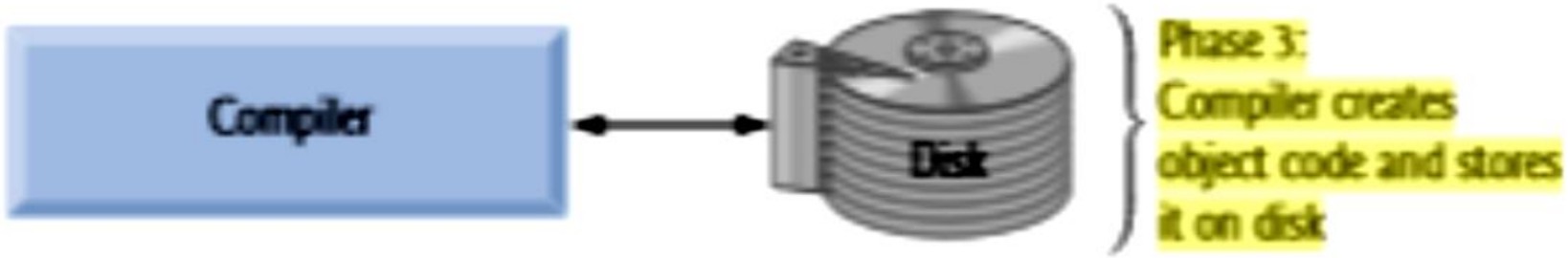
Phase 2: Preprocessing a C++ Program

- A preprocess program executes automatically before the Compiler's translation phase begins
- The C++preprocessor obeys commands called preprocessor directives, which indicate that certain manipulations are to be performed on the program before compilation.
- These manipulations usually include other text files to be compiled, and perform various text replacements.



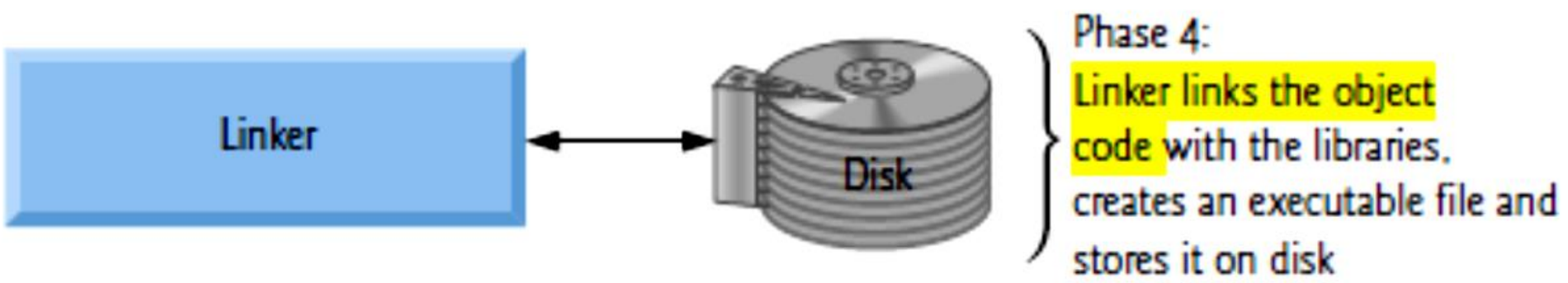
Phase3:Compiling a C++ Program

- In Phase 3, the compiler translates the C++ program into machine-language code—also referred to as object code.



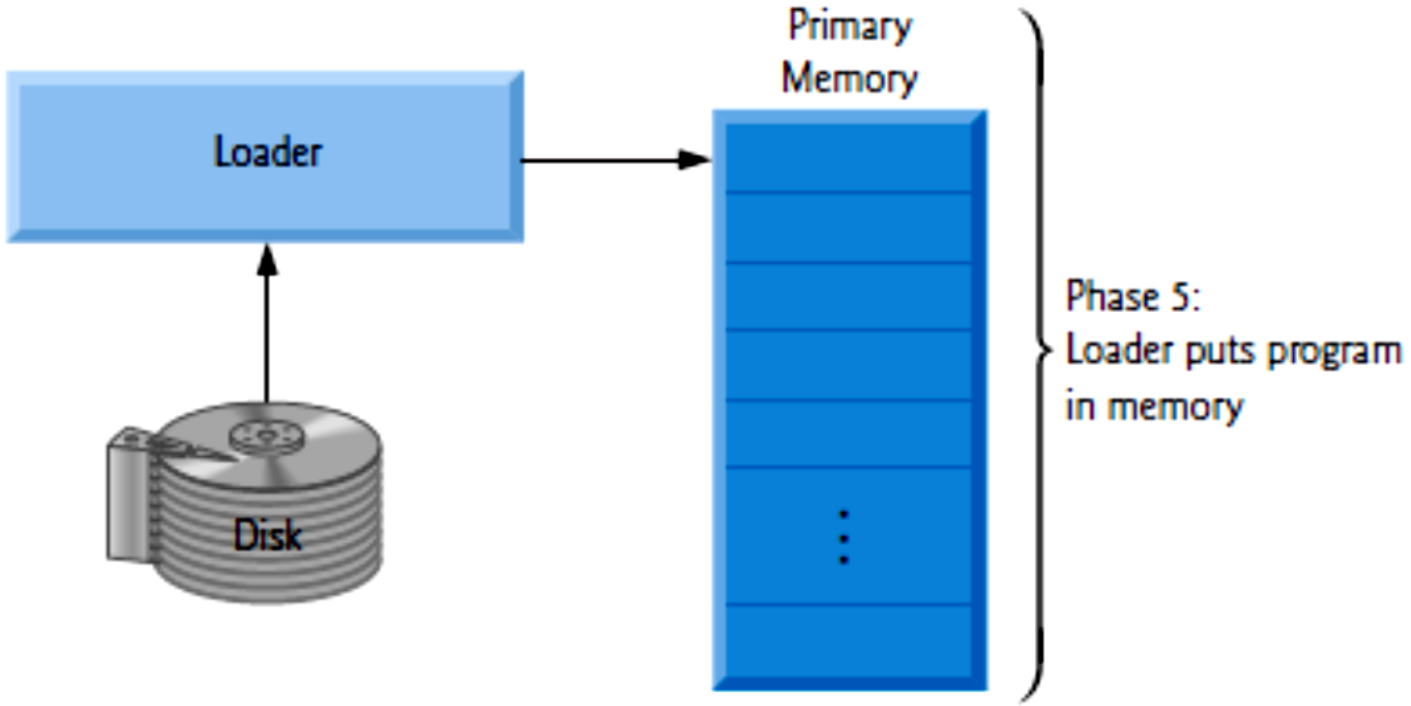
Phase4:Linking

- C++ programs typically contain references to functions and data defined elsewhere, such as in the standard libraries or in the private libraries.
- The object code produced by the C++ compiler typically contains “holes” due to these missing parts. A linker links the object code with the code for the missing functions to produce an executable program (with no missing pieces).
- If the program compiles and links correctly, an executable image is produced.



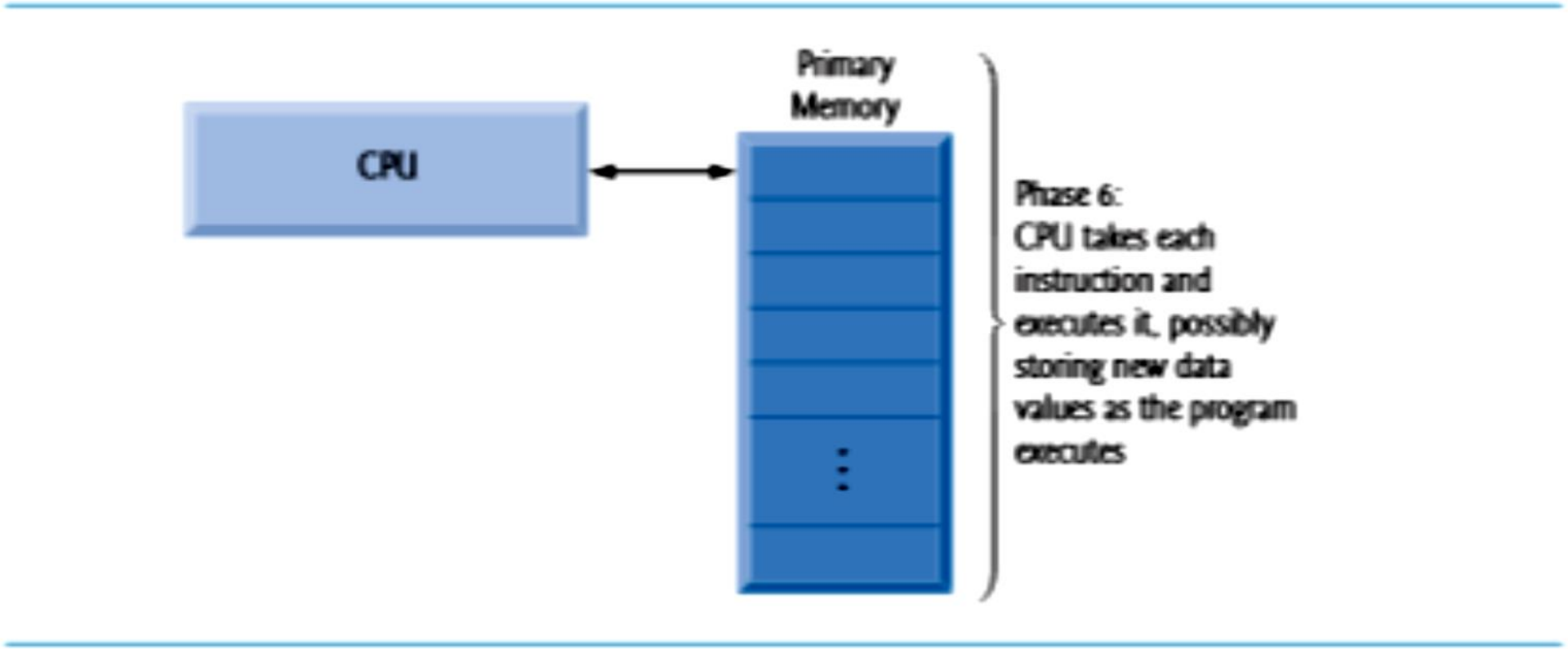
Phase5: Loading

- Before a program can be executed, it must first be placed in memory. This is done by the loader, which takes the executable image from disk and transfers it to memory.
- Additional components from shared libraries that support the program are also loaded.

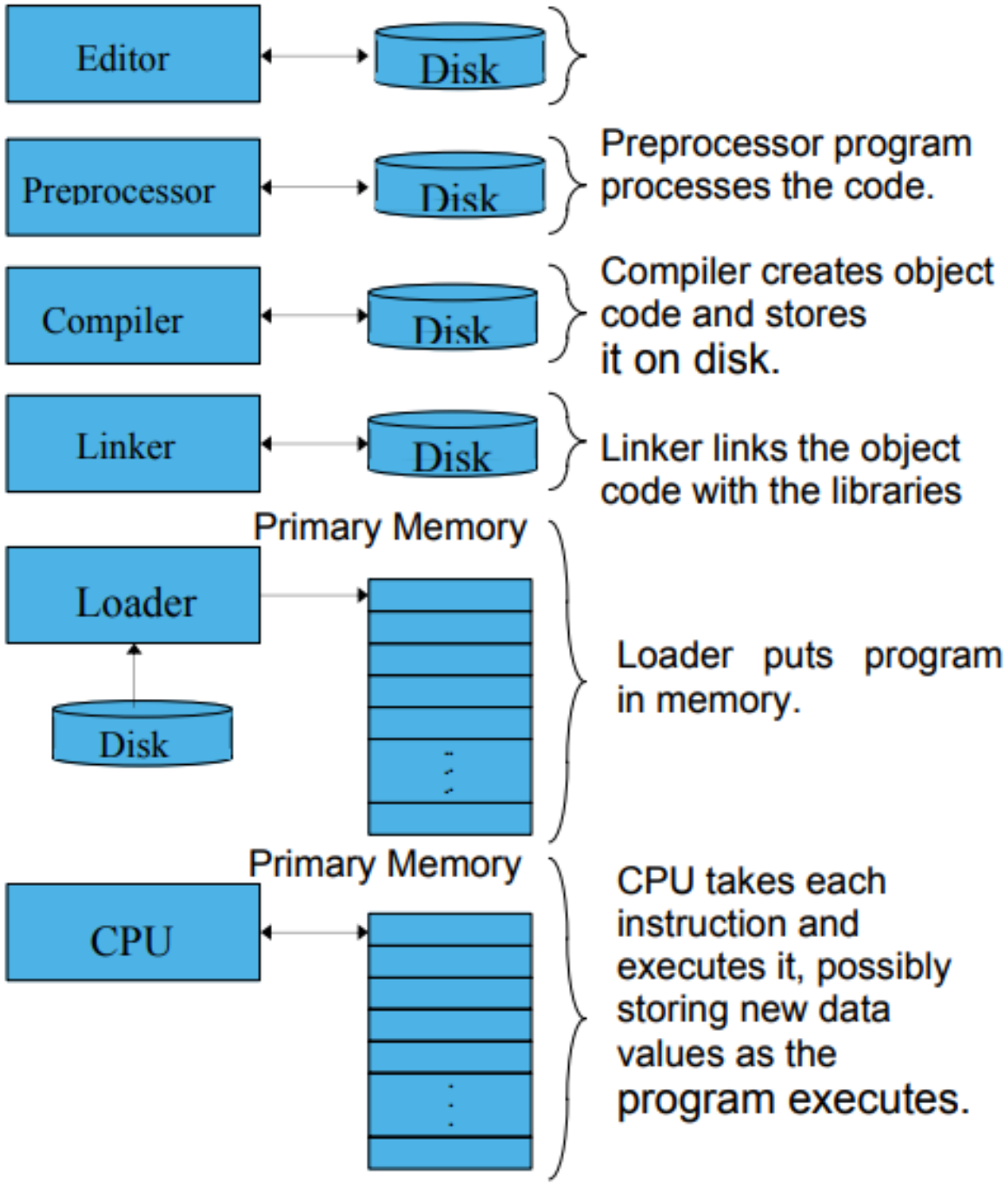


Phase6: Execution

- Finally, the computer, under the control of its CPU, executes the program one instruction at a time. Some modern computer architectures can execute several instructions in parallel.



Code Execution Cycle



#include Preprocessor Directive

- #include <iostream> is a preprocessor directive, which is a message to the C++ preprocessor.
- Lines that begin with # are processed by the preprocessor before the program is compiled
- This header must be included for any program that outputs data to the screen or inputs data from the keyboard using C++'s stream input/output.
- #include <iostream>
- Use #include <iostream.h> (older compilers)

Namespace

- A namespace is a collection of name definitions.
- One name, such as a function name, can be given different definitions in two namespaces.
- A program can then use one of these namespaces in one place and the other in another location.

- using std::cin; using std::cout; using std::endl;
- using namespace std;

Main Function

- int main() is a part of every C++ program. The parentheses after main indicate that main is a program building block called a function.
- Exactly one function in every program must be named main.
- It return an integer.

Console Input/Output

Output Using cout

- The values of variables as well as strings of text may be output to the screen using cout .Any combination of variables and strings can be output.
- cout << "Hello reader.\n"

Input Using cin

- Reads in a number that the user enters at the keyboard and sets the value of the variable.
- cin>>noOfItems

First C++ Program (1)

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

3  int main( )
4  {
5      int numberOfLanguages;

6      cout << "Hello reader.\n"
7           << "Welcome to C++.\n";

8      cout << "How many programming languages have you used? ";
9      cin >> numberOfLanguages;

10     if (numberOfLanguages < 1)
11         cout << "Read the preface. You may prefer\n"
12              << "a more elementary book by the same author.\n";
13     else
14         cout << "Enjoy the book.\n";

15     return 0;
16 }
```

First C++ Program(2)

Sample Dialogue 1

```
Hello reader.
Welcome to C++.
How many programming languages have you used? 0
Read the preface. You may prefer
a more elementary book by the same author.
```

← *User types in 0 on the keyboard. User input is shown in bold.*

Sample Dialogue 2

```
Hello reader.
Welcome to C++.
How many programming languages have you used? 1
Enjoy the book
```

← *User types in 1 on the keyboard. User input is shown in bold.*

Stream Insertion Operator

- The << operator is referred to as the stream insertion operator. When this program executes, the value to the operator's right, the right operand, is inserted in the output stream.
- (<<) inserts data into an output stream.

Stream Extraction Operator

- Reads in a number that the user enters at the keyboard and sets the value of the variable numberOfLanguages to this number.

- (>>) extracts data from an input stream.

Constant Modifier/Named Constants

- C++ provides a way of marking an initialized variable so that it cannot be changed. If your program tries to change one of these variables, it produces an error condition.
- To mark a variable declaration so that the value of the variable cannot be changed, precede the declaration with the word const (which is an abbreviation of constant).

```
const int BRANCH_COUNT = 10;
const int WINDOW_COUNT = 10;
```

Naming Constants with the const Modifier

```
const int BRANCH_COUNT = 10, WINDOW_COUNT = 10;
```

Named Constant (1)

```
1  #include <iostream>
2  using namespace std;
3
4  int main( )
5  {
6      const double RATE = 6.9;
7      double deposit

8      cout << "Enter the amount of your deposit $";
9      cin >> deposit;

10     double newBalance;
11     newBalance = deposit + deposit*(RATE/100);
12     cout << "In one year, that deposit will grow to\n"
13          << "$" << newBalance << " an amount worth waiting for.\n";

14     return 0;
15 }
```

Sample Dialogue

```
Enter the amount of your deposit $100
In one year, that deposit will grow to
$106.9 an amount worth waiting for.
```

```
1  //Program to demonstrate cin and cout with strings
2  #include <iostream>
3  #include <string>

4  using namespace std;
5  int main( )
6  {
7    string dogName;
8    int actualAge;
9    int humanAge;

10   cout << "How many years old is your dog?" << endl;
11   cin >> actualAge;
12   humanAge = actualAge * 7;

13   cout << "What is your dog's name?" << endl;
14   cin >> dogName;

15   cout << dogName << "'s age is approximately " <<
16          "equivalent to a " << humanAge << " year old human."
17          << endl;

18   return 0;
19 }
```

Needed to access the `string` *class.*

Using cin and cout with a String

Sample Dialogue 1

```
How many years old is your dog?
5
What is your dog's name?
Rex
Rex's age is approximately equivalent to a 35 year old human.
```

Sample Dialogue 2

```
How many years old is your dog?
10
What is your dog's name?
Mr. Bojangles
Mr.'s age is approximately equivalent to a 70 year old human.
```

"Bojangles" is not read into `dogName` *because* `cin` *stops input at the space.*