Lecture 1: Introduction

A modern beamer theme

Scott Day

October 13, 2016

Curtin University

Table of contents

Programming

What is a Computer Program?

A computer program is a collection of instructions that performs a specific task when executed by a computer. A computer requires programs to function, and typically executes the program's instructions in a central processing unit.

A part of a computer program that performs a well-defined task is known as an algorithm. A collection of computer programs, libraries and related data are referred to as software.

Problem Solving

Recipe to writing programs:

1. Understand the problem.
2. Think of a solution.
3. Describe the solution in as much detail as possible.
   You may use diagrams or plain English to do this.
4. Translate your solution into a program.
5. Run your program and see if it works.
   - Yes? Hurray! Victory!
   - No? Go back to 1

Example

Think like a computer!

What steps do you need to take to draw a smiley face?

Example

C++

The C++ Programming Language

C++ is a general-purpose programming language with a bias towards systems programming that:

- Is a better C,
- Supports data abstraction,
- Supports object-oriented programming, and
- Supports generic programming.

The Pervasiveness of C++

Why choose C++

Brief History of C++

The C++ programming language is an extension of C that was developed by (the god himself) Bjarne Stroustrup in the early 1980s at Bell Laboratories.

Example C++ Program (Hello World)

```cpp
#include <iostream>

main()
{
    cout << "Hello World!";
    return 0;
}
```

Output: Hello World!

Environment Setup

Setup and Installation

C++ Editors

Compiling

Editing, Compiling, and Execution

A Simple Program to Add Two Numbers

The following is an example of a simple program written in C++. The program is designed to read two numbers typed by a user at the keyboard; compute their sum and display the result on the screen.

```
1  // Program to add two integers typed by user at keyboard
2  #include <iostream>
3  using namespae std;
4
5  int main()
6  {
7      int num1, num2, total;
8
9      cout << "Enter integers to be added:" << endl;
10     cin >> num1 >> num2;
11     total = num1 + num2;
12     cout << "The sum is " << total << endl;
13
14     return 0;
15 }
```

Program Structure and Syntax

C++ uses notation that may appear strange to non-programmers (and me).
The notation is part of the programming language `syntax`.

Syntax Formal rules that specify the structure of a legal program.

the notation and explanations which follow will appear strange if you have never written a computer program.

Don’t worry about them or how the program works. This will be explained in more detail later.

The following is an overview.

Program Structure and Syntax

Every C++ program consists of a header and a main body and has the following structure:

```
1  // Comment statements which are ignored by computer
2  /* Also a comment */
3  #include < header file name >
4
5  int main()
6  {
7      declaration of variables;
8      statements;
9
10     return 0;
11 }
```

Program Structure and Syntax

```
1  // Program to add two integers typed by user at keyboard
2  #include <iostream>
3  using namespae std;
4
5  int main()
6  {
7      int num1, num2, total;
8
9      cout << "Enter integers to be added:" << endl;
10     cin >> num1 >> num2;
11     total = num1 + num2;
12     cout << "The sum is " << total << endl;
13
14     return 0;
15 }
```

Line 1

- Lines beginning with // indicate that the rest of the line is a **comment**.
- Comments are inserted by programmers to help people read and understand the program.
- Can be placed anywhere in a program.

Program Structure and Syntax

```
1  // Program to add two integers typed by user at keyboard
2  #include <iostream>
3  using namespae std;
4
5  int main()
6  {
7      int num1, num2, total;
8
9      cout << "Enter integers to be added:" << endl;
10     cin >> num1 >> num2;
11     total = num1 + num2;
12     cout << "The sum is " << total << endl;
13
14     return 0;
15 }
```

Line 2

- Lines beginning with # are instructions to the compiler's preprocessor.
- The **include** instruction says "what follows is a file name, find that file and insert its contents right here".
- Here the file iostream contains the definitions of **cin**, **cout**.

Program Structure and Syntax

```
1  // Program to add two integers typed by user at keyboard
2  #include <iostream>
3  using namespae std;
4
5  int main()
6  {
7      int num1, num2, total;
8
9      cout << "Enter integers to be added:" << endl;
10     cin >> num1 >> num2;
11     total = num1 + num2;
12     cout << "The sum is " << total << endl;
13
14     return 0;
15 }
```

Line 3

- Specifies that names used in the program (ie. **cin** and **cout**) are defined in the standard libraries.
- This is used to avoid problems with other libraries which may also use these names.

Line 5

Development Environment & Development Cycle