SANTA MONICA COLLEGE

CS 50 : C Programming

Jinan Darwiche

SANTA MONICA COLLEGE

Topics

- Welcome to class
- Computer languages
- The compilation process·
- The history of C
- Developing programs with C
- Helloworld.Cpp

Welcome to Class

- Each unit contains a slides file, reading materials, an assignment or a quiz, and Discussions
- The slides are your starting point of the learning process
- Check the due dates of the assignments, quizzes and or Discussion using our course site
- Ask questions in the Discussion if you have them; you may also answer others' questions – The Discussion is our "meeting" place. Check it daily

Programming: Why do it?

- Processor based devices such as desktops/laptops, “smart” phones, electronic pads, even some cars, TV’s, etc execute instructions, called machine language software.
- Instructions are written in a programming language of your choice. This is called Source Code.
- Another software, called Compiler, changes the source code into Assembly/Machine Code.

Computer languages

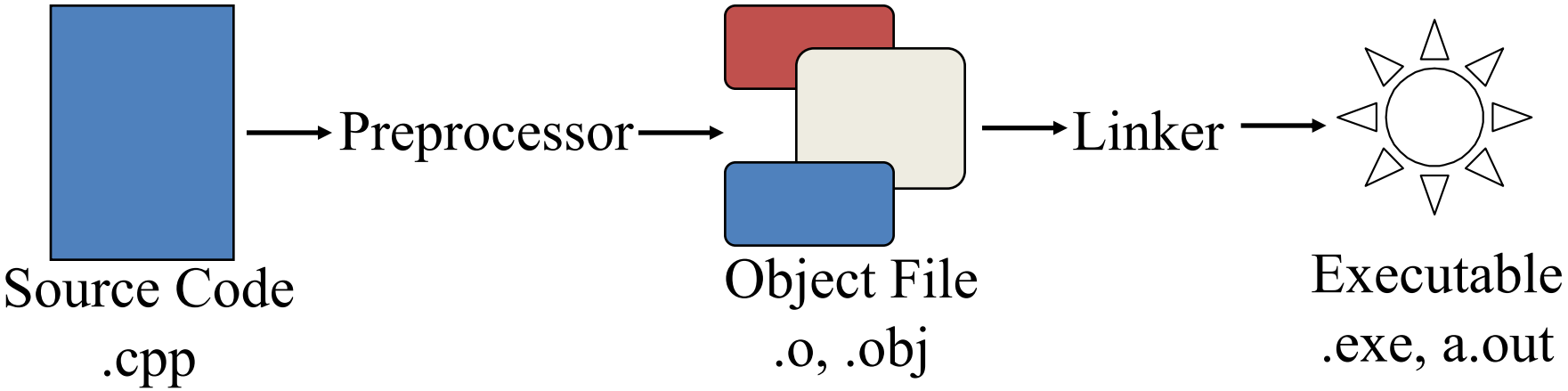
- Computer languages have evolved over time
- Initially, programmers coded in machine language : 01010110 0001000
- Eventually, assemblers were made to hide machine language behind mnemonic instruction: ADD R1, 8

High-Level Languages

- C offers convenient “high-level” language features with access to low-level hardware primitives
- Languages are interpreted or compiled
- C is a compiled language

Compiled Languages

- Compiled languages must be turned into executable computer instructions



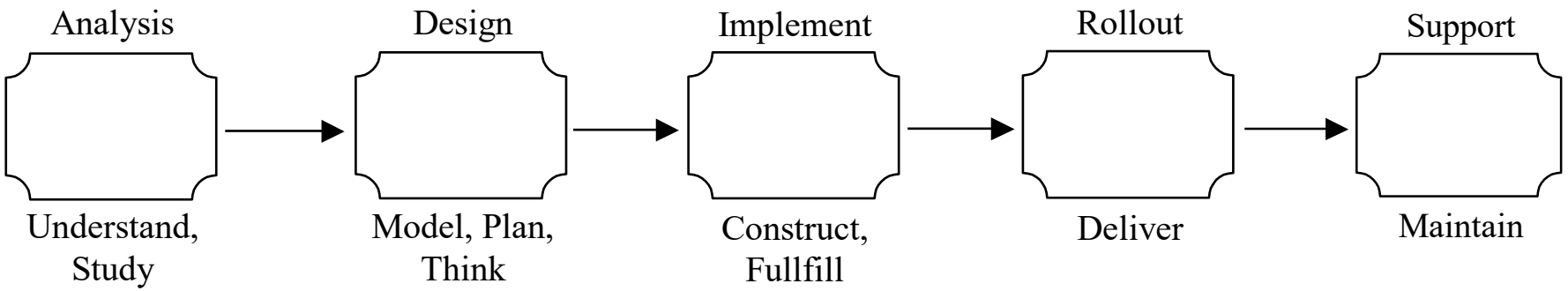
- Errors can occur at each step!
  - Compile-time, linkage, run-time

The History of C

- Authored by Dennis Ritchie, AT&T bell labs
- Originally, C was a programming tool to help in creating the UNIX operating system for the DEC PDP-11 computer
- Language is now an international standard

Developing Programs With C

- Program Development Methodology



- Our initial programs won't require much analysis or design

Visual Studio .NET

- We will use the Visual Studio compiler
- Visual Studio .NET, let's call it VS contains several compilers, including C++.
- C++ is a language that supersedes C. We can use it to write and compile C code.
- If you have not done so already, follow the instructions to install VS on your machine. Other compiler versions are NOT accepted.

Purpose of Software

- Most of the software/programs we will learn and write will follow the same pattern:
  - Get input
  - Process the input to product some result
  - Show output – usually the result produced in the previous step

To Learn Programming You Learn

- The **syntax**: What words and how to put them together to form sentences
  - Words are referred to as **keywords**, and sentences are referred to as **statements**
  - **Each Programming language has its own syntax**
- The **semantics:** How to form statements to solve a particular problem.
  - Semantics are universal – not language dependent

Learning Programming

- Analyzing existing code helps in understanding how to write code
- In the early stages, you may not understand every line – focus on the main concepts
- Syntax errors are very common – be patient

Time For Our First Demo: Hello World

- Follow the video shown in the module.

What is Hello World

- A starting point to learn how to show output.
- Code is typically made of one statement to show the words “hello world” as output.

Write the Code

```
#include "stdio.h"

int main(void)
{
printf("Hello World!");
return 0;
}
```

Understanding the Code

#include "stdio.h” ← *this line is needed so we can show output and get input - Lines with # go together at the top before the int main() line*

← *You can leave blank lines to make the code more readable. This is called white space*

int main(void) ← *this is the starting point of executing the code*

{ ← *after main, you must enclose your code inside curled brackets. This is called code block*

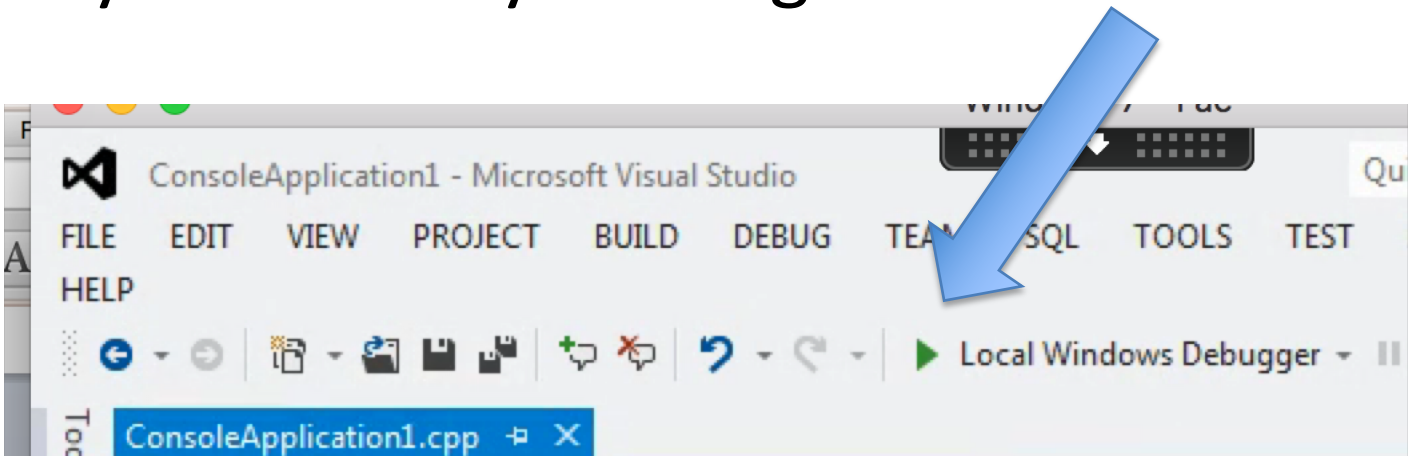
printf("Hello World!"); ← *this is a statement – every statement must end with a semi-colon*

return 0; ← *every function must "return" a value. We will discuss this in more details*

} ← *every opened curled bracket must be closed with a closing curled bracket.*

After you type the code

- Click the Debug button (green play icon)
- You must always test if your code is syntax error free and executes without errors.
- If you get an error, you must fix it
- Run your code by clicking:



When you run….

- The code we covered for Hello World, runs, but you don’t get to “see” it
- The runtime Console screen appears then quickly disappears
- To make the Console stay, we need an input statement
- After the top { add:
  int temp;
- After the printf line, add:
  scanf_s(”%d”,&temp);

C Code components

- #Include directives (they are not called statements)
- main function
- Statements inside functions
- Comments: lines that explain the code but the compiler does not translate to machine language.
- Comments can be single-line: // one line comment or extend over multiple lines using /* this is a multi line comment */
- Comments help programmers understand the code later when they read it

Output in C

- C output statement:
  printf(“Hello World \n”);
  - sends information from program to the screen - this is what VS refers to as the Console(standard output)
  - double quotes "..." delimit a string
  - \n sends a new-line-character – this is optional

Overview of C and Programming

- Most programs get input from the user
- Input in code is saved into Variables
- Input is processed to produce some result
- Results are also saved in variables
- A variable is a location in RAM, marked by its name, and what type of information it can contain.

Next Module

- We learn how to get input, process it then show output
- This requires the use of variables, input and output statements
- Review these slides again after reading the book and watching the videos
- Post questions in the Discussion
- Complete the quiz/assignment