Getting Started in C

CSC230: C and Software Tools

N.C. State Department of Computer Science

Computer Science NC STATE UNIVERSITY

Outline

- C Overview
- Software Tools
- Course Goals
- Programming Languages
- Common Platform
- Sample Program
  - C Example
  - Java Example

Computer Science NC STATE UNIVERSITY

Why Yet Another Programming Course?

- Learn a widely-used programming language that is procedural (not object-oriented)
- Help you transition from basic programming to…
  - Operating Systems (CSC246)
  - Software Engineering (CSC326)

Computer Science NC STATE UNIVERSITY

Why C?

- Developed to build Unix operating system
- Main design considerations:
  - Compiler size: needed to run on PDP-11 with 24KB of memory (Algol60 was too big to fit)
  - Code size: needed to implement the whole OS and applications with little memory
  - Performance
  - Portability
- Little (if any consideration):
  - Security, robustness, maintainability
  - Legacy Code

Computer Science NC STATE UNIVERSITY

Why C? (con't)

- Simple to write compiler
  - Programming embedded systems, often only have a C compiler
- Performance
  - Typically 50x faster than interpreted Java
- Smaller, simpler, lots of experience
- One of the most popular programming languages
  - See http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html for the latest numbers

What's Your Priority?

Priority	Language Choices
Speed of execution, minimum memory "footprint"	Assembly, C
Safer, easier to develop large (hundreds of files) programs	Java, C++
Easier / faster to code, higher level operations, richer libraries	Python, Ruby, PHP, Perl
Integrate with the web	Web application frameworks, Javascript

C Strengths

- It's a procedural language (like many others)
- It's efficient (binary code size, execution speed)
- Simple, clean language design
- C99 is a international standard
- It has a decent standard library of useful functions

Computer Science NC STATE UNIVERSITY

Examples of C or C++

- Linux: Assembly, C
- MS Windows: Assembly, C, C++
- Firefox Web Browser: C++, Javascript
- Gnu Compiler (GCC): C
- MySQL: C, C++
- Embedded Systems (cars, appliances, etc.)
- High performance (science/engineering) applications

Computer Science NC STATE UNIVERSITY

C Weaknesses

- Little consideration for security or safety
- Less modular than Java and other OO languages (but C++ fixes that)
- More programming effort required than PHP/Python/Perl/Ruby and other scripting languages
- Not usually written in C or C++: web apps, business apps, GUIs, simple utility programs

Software Tools

- Help produce programs…
  - quickly
  - of high quality
  - In large teams of programmers
- Examples of tools
  - Compilers, code formatters / indenters, debuggers, test generators, performance profilers, version control management, dependency checking, documentation generation, static analysis, …
- Often these are bundled in an IDE
  - Eclipse, Visual Studio, …

Some Standard Goals

- Understand syntax and semantics of C and how to use
- Be able to write small- to medium-sized C programs
- Understand differences between compiling and interpreting
- Know how to avoid, find, and fix programming bugs in C
- Know how to dynamically allocate/free memory
- Know how to use header files and the C preprocessor
- Be familiar with and know how to use standard library functions
- Use command-line tools to design, compile, document, debug, improve, and maintain programs
- Know how to automate dependence checking / building an executable / common programming tasks
- Know how to use common tools to write programs as part of a team

Computer Science NC STATE UNIVERSITY

Other Goals

- Will this course make me…
  - ✓ a better programmer?
  - ✓ a better computer scientist?
  - ✓ more marketable?
  - ? wealthy, successful, famous?
  - ? a better person?

Computer Science NC STATE UNIVERSITY

Getting Started….

Computer Science NC STATE UNIVERSITY

Types of Programming Languages

- *Declarative*: focus on what the computer should do
  - *Functional:* Scheme, Haskal
  - *Dataflow*
  - *Logic-* or *constraint-based: Prolog*
  - *Markup languages:* HTML, CSS, subset of SQL
- *Imperative*: focus on how the computer should do something
  - *Procedural* : C
  - *Object-oriented* : Java

Computer Science NC STATE UNIVERSITY

Procedural vs. O-O

- Procedural: programming as procedures that modify variables
  - emphasis on actions that must take place
  - analogy: following a recipe
- O-O: programming as objects that interact (each with internal state, and methods to manage that state)
  - emphasis on the state of objects
  - analogy: operating a car

Common Platform for This Course

- Different platforms have different conventions for end of line, end of file, tabs, compiler output, …
- Solution (for this class): compile and run all programs consistently on one platform
- Our common platform:

Intel PC + Linux

Your Choices

Option	Use GUI-based Editor?	Access to your unity Filespace?
Use Unity Lab Computer	Y	Y
ssh to VCL (linux)	N**	Y
ssh to remote-linux.eos.ncsu.edu	N**	Y
Use Mac OS X (+developer tools)	Y	ftp*
Use MS Windows + cygwin	Y	ftp*
Use Linux on your PC (dual boot or virtualized)	Y	ftp*

- * direct if you install realm kit
- ** Yes if you run X windows server on your computer

Computer Science NC STATE UNIVERSITY

Common Platform Questions

- If you want to develop locally, that's fine, but you must ensure that it works on the Common Platform
  - You should always test on the Common Platform before submitting
  - No, really, you should test on the Common Platform
  - There are differences between the C compilers for different architectures that may cause your program (that runs locally) to fail on the Common Platform
  - C is not architecture neutral!

Computer Science NC STATE UNIVERSITY

The "Zeroth" C Program

```
#include <stdio.h>

int main ()
{
    printf("Hello, world!\n");
    return 0;
}
```

File with library function declarations

Entry point of the program, with no arguments

Exit program and indicate successful completion

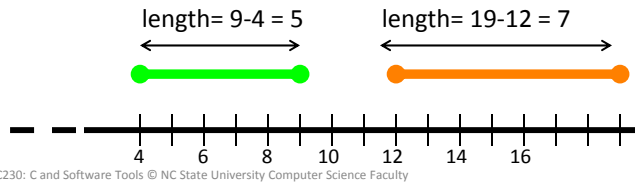
Standard library function, with message argument

```
% gcc –Wall –std=c99 hello.c –o hello
```


Computer Science NC STATE UNIVERSITY

A Simple Program (For Illustration)

- Specification
  1. Two line segments are created
  2. The user is requested to enter the left and right edges of the two line segments, as integer values
  3. The length of each segment is computed as (right edge – left edge)
  4. The two lengths are compared to determine if they are the same, and a message is displayed



Computer Science NC STATE UNIVERSITY

Compiling and Running the Program

Terminal — bash —

```
cmd> gcc -Wall -std=c99 intro.c -o intro
cmd> intro
Enter left edge of segment 1: 5
Enter right edge of segment 1: 10
Enter left edge of segment 2: 20
Enter right edge of segment 2: 30
segment lengths are NOT equal
cmd>
```


The First C Program (part 1)

```
#include <stdio.h>
#include <stdlib.h>

static int  compute_length (int, int);

int main (void)
{
    typedef struct {
        int         left;
        int         right;
        int         length;
    } seg_t;

    seg_t *seg1, *seg2;
```

library function definitions

main routine, procedure #1

data structure definition

declaration of references to data structure instances

First C Program (part 2)

```
    seg1 = (seg_t *) malloc (sizeof (seg_t));
    seg2 = (seg_t *) malloc (sizeof (seg_t));

    printf ("Enter left edge of segment 1: ");
    scanf ("%d", &(seg1->left));
    printf ("Enter right edge of segment 1:");
    scanf ("%d", &(seg1->right));
    printf ("Enter left edge of segment 2: ");
    scanf ("%d", &(seg2->left));
    printf ("Enter right edge of segment 2:");
    scanf ("%d", &(seg2->right));

    seg1->length = computelength (seg1->left,
         seg1->right);
    seg2->length = computelength (seg2->left,
         seg2->right);
```

create instances of data structure, and associate with references

input / output, store result in data structure

call a subroutine, store result in data structure

First C Program (part 3)

```
  if (seg1->length == seg2->length)
    printf("Segment lengths are equal\n");
  else
    printf("Segment lengths are NOT equal\n");

    return 0;
}

int compute_length (int left, int right)
{
    return (right-left);
}
```

subroutine, procedure #2

Same Program, in Java (part 1)

```
import java.util.Scanner;

public class Segment {
   private int left;
   private int right;
   private int length;

   public Segment(int l, int r) {
      left = l;
      right = r;
   }
   public int computeLength( ) {
      length = right - left;
      return length;
   }
```

Class definition

Object state

Object constructor

Object behavior

Same Program, in Java (part 2)

```
public static void main(String [ ] args ) {
   Scanner in = new Scanner(System.in);
   System.out.print("Enter left edge of segment 1: ");
   int l = in.nextInt();
   System.out.print("Enter right edge of segment 1: ");
   int r = in.nextInt();

   Segment seg1 = new Segment (l,r);

   System.out.print("Enter left edge of segment 2: ");
   l = in.nextInt();
   System.out.print("Enter right edge of segment 2: ");
   r = in.nextInt();

   Segment seg2 =  new Segment (l,r);
```

Read in data

Create object

Rinse and repeat for Seg 2

Same Program, in Java (part 3)

```
 int len1 = seg1.computeLength( );
 int len2 = seg2.computeLength( );

 if (len2 == len1)
  System.out.println("segment lengths are equal\n");
 else
  System.out.println("segment lengths are NOT equal\n");
}
}
```

Finish calculation

Some Quotes About C

- "The major problem with new programmers is that they don't have experience in languages that require them to do manual memory management." – Bill Gates
- "C is quirky, flawed and an enormous success." - Dennis Ritchie
- "I view the landslide of C use in education as something of a calamity." - Nicklaus Wirth

Quotes… (cont'd)

- "Unix and C are the ultimate computer viruses." - Richard P Gabriel
- "C is often described … as a language that combines all the elegance and power of assembly language with all the readability and maintainability of assembly language." - MIT Jargon Dictionary

Exercise 01.02

- Watch the video "Common Platform"
  - Linked in under the top level topic for the course
- After watching the video, fill out the Google Form titled "Exercise 01.02"
- Due 10 minutes before the next class period