Programming Studio #7

ECE 190

Programming Studio #7

- Administrivia
- Concepts this week:
  - Introduction to C programming
  - Control constructs
  - Standard output (`printf`)
- Exercise: program to execute 4 basic math functions

Announcements

- Exam 1 returned
  - Regrade requests on separate sheet of paper
  - Regrade requests due next Friday
- MP2.2 due Wednesday (3/10) by 5p (Start now)

Introduction to C Programming

- C is a high-level language – Can give symbolic names to values
- Better expressiveness, abstraction, readability
- Easier to code
  - Machine code: 0001 001 001 1 00001
  - Assembly: ADD R1, R1, #1
  - High-level: R1 = R1 + 1

Conditional Constructs

```
#include <stdio.h>
#define A 4
int main(){
   printf("A is %d",A);
   if (A>0) {printf("A is pos");}
   else if (A<0) {printf("A is neg");}
   else {printf("A is zero");}
}
```

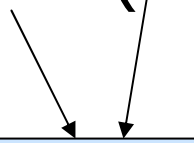
Get `first.c` from web as example

```
if (condition)
{<code>;}else {}
```

- See Appendix D.7
- Like branch (BR) – changes flow of execution

Standard Output

```
#include <stdio.h>
#define A 4
int main(){
   printf(“A is %d”,A);
   if (A>0) {printf(“A is pos”);}
   else if (A<0) {printf(“A is neg”);}
   else {printf(“A is zero”);}
}
```



```
printf(“<string>”, args);
```

- See `man 3 printf`
- Can format output (e.g., numbers)

`printf` Conversion Specifiers

- Integers (e.g., int)
  - `printf("Count = %d", 5);`
  - `printf("Count = %d", count);`
- Floating point (e.g., float)
  - `printf("Pi = %f", 3.14159);`
  - `printf("Pi = %f", pi);`
- Strings
  - `printf("Hello, my name is %s", name);`
- See Appendix D.9

Compiling and Executing

- Use gcc (GNU C Compiler)
- Download “first.c”
- Compile: `gcc first.c –o first.out`
  - Syntax:
    `gcc <C file> -o <executable>`
- Execute: `./first.out`

Programming Assignment

- Write a program that can add, subtract, multiply, and divide 2 numbers.
- 2 numbers and operation to be performed defined using `#define` directives
- Operation to be performed is defined in the table

Operation requested	Operation performed
1	Addition
2	Subtraction
3	Multiplication
4	Division

Multiply Ex (Ad Nauseam) (1)

```
/* Multiple two numbers */

#include <stdio.h>
#define A 4

int main(){
   int b,c;
   b = 10;
   c = A * b;
   printf("A mult B = %d", c);
}
```

Comment

Preprocessor directives

Main function

Multiply Ex (Ad Nauseam) (2)

```
/* Multiple two numbers */

#include <stdio.h>
#define A 4

int main(){
   int b,c;
   b = 10;
   c = A * b;
   printf("A mult B = %d", c);
}
```

Variable ***declaration***:
- See Appendices D.3-D.4
- Syntax: <type> <name>
- Example types
  - char (8 bits)
  - int (32 bits)
  - float (32 bits)
  - double (64 bits)

Assignment

Different?

Multiply Ex (Ad Nauseam) (3)

```
/* Multiple two numbers */

#include <stdio.h>
#define A 4

int main(){
   int b,c;
   b = 10;
   c = A * b;
   printf("A mult B = %d", c);
}
```

Operators:
- Lots (see Appendix D.5)
- Example arithmetic
  - `+` (add)
  - `-` (subtract)
  - `*` (multiply)
  - `/` (divide)

Function call (for output)