Computer Science 5C
Chapter 1

Dr. Scott A. Brandt
Professor, Computer Science Department
Director, UCSC/LANL Institute for Scalable Scientific Data Management

CMPS 5C - 1/7/08
What is a program

• A recipe
• A list of computer instructions
• A set of steps for the computer to take
The programming process

• Specify a task
• Discover an algorithm for its solution
• Code the algorithm (in C)
• Compile the program
• Test the code
Software development process

• Design
• Edit
• Compile
• Execute
• Lather, rinse, repeat
A first program

/* The traditional first program in honor of Dennis Ritchie, inventor of C at Bell Labs in 1972 */

#include <stdio.h>

int main(void)
{
    printf("Hello, world\n");
    return 0;
}
/* The traditional first program in honor of Dennis Ritchie, inventor of C at Bell Labs in 1972 */

#include <stdio.h>

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

Variables, Expressions, and Assignments

• Convert depth (7 fathoms) to other units

• Algorithm
  • Assign the number of fathoms to a variable
  • Convert fathoms to feet and store in a variable
  • Convert feet to inches and store in a variable
  • Print the different units
Hesperus

#include <stdio.h>

int main(void) {
    int inches, feet, fathoms;
    fathoms = 7;
    feet = 6 * fathoms;
    inches = 12 * feet;
    printf("Wreck of the Hesperus\n");
    printf("Its depth at sea in different units:\n");
    printf(" %d fathoms\n", fathoms);
    printf(" %d feet\n", feet);
    printf(" %d inches\n", inches);
}

Anatomy of a program

preprocessing directives

int main(void)
{
    declarations
    statements
}
Arithmetic operators

• +
• -
• *
• /
• %

• Binary operators: a + b, a - b, a * b, a / b, a % b
• Assignment: =
• c = a + b
Declaring and using variables

```c
#include <stdio.h>

int main(void)
{
    char c, d;

    c = 'A'; d = 'W';
    printf("%c klajshdfasdfs %c\n", c, 'W');
    return 0;
}
```
Mathematical variables

```c
#include <stdio.h>

int main(void)
{
    float x, y;
    x = 1.5;
    y = 2.7;
    printf("The sum of x and y is %f\n", x + y);
    return 0;
}
```
Assignment

• $a = b + c$;
• $z = 5$;
• $x = 2 + b$;

• NOT: $a + b = c$;
• NOT: $5 = z$;
• NOT: $x + 2 = b$;
Initialization

• It is possible to declare a variable and assign it a value at the same time

• char c = ‘A’;
• int x = 7;
• int y = x;
Initialization

```c
#include <stdio.h>

int main(void)
{
    int inches, feet, fathoms = 7;
    feet = 6 * fathoms;
    //<etc.>

    OR

    int inches, feet, fathoms = 3 + 4;

    BUT NOT

    int inches, feet = 6 * fathoms, fathoms = 7;
```
#include

- Tells the compiler to include the named file
- `#include <stdio.h>`
- `#include <filename>`
  - Look in system directories for the file
- `#include "filename"`
  - Look in the current directory for the file
#define

• Can be used to define static variables
• #define PI 3.14159
• #define SECONDS_PER_MINUTE 60
• #define UPPER_LIMIT 100
• #define UPPER_LIMIT 100000
printf() and scanf()

• printf() is used to print formatted output on the screen
• printf(<control_string>, <arguments>);

• printf(“Hi\n”);
• printf(“The answer is: %d”, answer);
• printf(“%f %f %f\n”, one, two, three);
printf() conversion characters

- c -- character
- d -- decimal integer
- e -- floating point number in scientific notation
- f -- floating point number
- g -- in the shorter of e and f
- s -- string of characters
Examples

• `printf("Get set: %s, %d, %f, %c%c\n", "one", 2, 3.33, ‘G’, ‘0’);`

• `printf("%c%3c%7c", ‘A’, ‘B’, ‘C’);`

• `printf("Some numbrs: %.1f %.2f %.3f\n", 1.0, 2.0, 3.0);`

• `printf("More numbers: %7.1f%72.f%7.3f\n", 4.0, 5.0, 6.0);`
scanf()

- Used for formatted input
- `scanf("%d", &x);`
- `scanf("%c", &c);`
scanf() conversion characters

- c  -- character
- d  -- decimal integer
- f  -- floating point integer
- lf -- long floating point integer
- Lf -- really long floating point integer
- s  -- string
Using printf() and scanf()

```c
#include <stdio.h>

int main(void)
{
    char first, middle, last;
    int age;

    printf("Input your three initials and age: ");
    scanf("%c%c%c%d", &first, &middle, &last, &age);
    printf("Greetings %c.%c.%c. %s %d.\n", first, middle, last, "Next year your age will be ", age+1);
}
```
Another simple program

```c
#include <stdio.h>
#define PI 3.141592653589793

int main(void)
{
    double radius;

    printf("%s

%s",
    "This program computes the area of a circle.", "Input the radius: \\
    ");
    scanf("%lf", &radius);
    printf("%s
%s\%.2f%s\%.2f
%s\%.5f\n\n",
    "Area = PI * radius * radius",
    " = ", PI, " * ", radius, " * ", radius,
    " = ", PI * radius * radius);

    return 0;
}
```
The *while* statement

```c
#include <stdio.h>

int main(void)
{
    int i = 1, sum = 10;

    while (i < 10) {
        sum = sum + 1;
        i = i + 1;
    }

    printf("Sum = %d\n", sum);
    return 0;
}
```
Compute sums

#include <stdio.h>

int main(void)
{
    int cnt = 0;
    float sum = 0.0, x;

    printf("The sum of your numbers will be computed\n\n");
    printf("Input some numbers: ");
    while(scanf("%f", &x) == 1) {
        cnt = cnt + 1;
        sum = sum + x;
    }

    printf("\n%s%5d
%s%12f\n\n", "Count:", cnt, "   Sum: ", sum);
    return 0;
}
Coding style

- Naming
- Indentation
- Spacing
- Comments