Data and C

COEN 10

C -- Lecture 2

Data and C

★Data are the values in the program

Input → Manipulate → Output

Input/Output

★Output

◎Function printf

printf ("string with placeholders", variables to replace placeholders);

◎Example

printf ("x = %d\n", x);

Input/Output

★Input

◎Function scanf

scanf ("string with placeholders", address of variables to receive values);

◎Example

scanf ("%d", &x);

Data Variables and Constants

- ★Constants
  - ◎Preset data
  - ◎never change values
- ★Variables
  - ◎Change or are assigned values as the program runs

Data: Data-Type Keywords

- ★Data types
  - ◎Used to identify the type of a variable

Data: Data-Type Keywords

- ★Keywords
  - ◎Original
    - ✧int, long, short, unsigned, char float, double
  - ◎C90
    - ✧signed, void
  - ◎C99:
    - ✧_Bool, _Complex, _Imaginary

Data: Data-Type Keywords

- ★Two families
  - ◎Integer types
    - ✧Number with no fractional part
    - ✧Stored as binary numbers
  - ◎Floating point types
    - ✧Number with a fractional part
    - ✧Not stored as a binary number

Data: Data-Type Keywords

★Integers: Binary representation

◎A value is represented as a binary number

0^{th} bit * 2^0 +

1^{st} bit * 2^1 +

2^{nd} bit * 2^2 + …

Data: Data-Type Keywords

★Floating point representation

◎A value is represented as a combination of fields

Exponent - base 10

Fraction - value between 0 and 1

Sign - + or -

Basic C Data Types

★int

◎Signed integer

◎Size depends on the computer system

✧16, 32, 64 bits

◎Constants

✧Numbers with no decimal point that fit into an integer

Basic C Data Types

★int

◎Declaring

```
int  x;
int  y = 5;
```

◎Initializing

```
x = 0;
```

Basic C Data Types

★int

◎Printing

```
printf ("x = %d", x);
```

◎Reading from keyboard

```
scanf ("%d", &x);
```

Basic C Data Types

★int

◎Other integer types

- ✧unsigned
- ✧short and unsigned short
- ✧long and unsigned long
- ✧long long and unsigned long long

Basic C Data Types

★char

◎1 byte -- values from 0 to 127

◎Typically holds the value of a character in ASCII code

- ✧'0' - '9' -- 48-57
- ✧'a' - 'z' -- 97-122
- ✧'A' - 'Z' -- 65-90

Basic C Data Types

★char

◎Constants

- ✧Use single quotes or the ASCII value

Basic C Data Types

★char

◎Declaring

```
char c;
char d = 'a';
char e = 65;  //ok, but bad style
```

◎Initializing

```
c = 'z';
```

Basic C Data Types

★char

◎Non-printing characters

✧use the ASCII code

```
char beep = 7;
```

✧or a escape sequence

\n - newline

\t - tab

\\ - backslash

Basic C Data Types

★char

◎Printing

```
c = 'A';
printf ("The letter is %c.\n", c);
printf ("The ascii code is %d.\n", c);
```

◎Reading from keyboard

```
scanf ("%c", &c);
```

Basic C Data Types

★floating point

◎Real numbers represented in scientific notation

◎Constants

✧Numbers with a decimal point or in scientific notation

Basic C Data Types

★floating point

◎Declaring

```
float f;
float g = 0.5;
```

◎Initializing

```
f = 2.87e-3
```

Basic C Data Types

★floating point

◎Printing

```
printf ("%f can also be %e\n", f, f);
```

◎Reading from keyboard

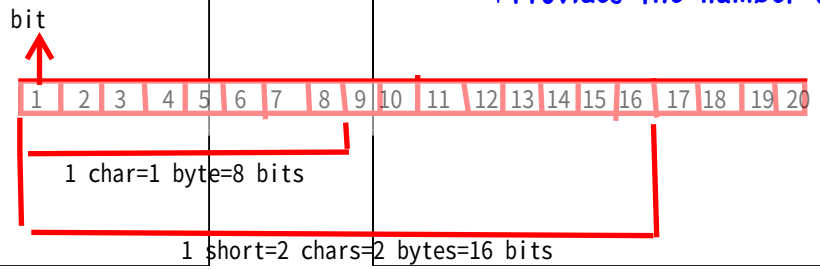
```
scanf ("%f", &f);
```

Basic C Data Types

★Type sizes

◎Linux and OS X

- ✧char - 8
- ✧int - 32
- ✧short - 16
- ✧long - 32
- ✧long long 64



Basic C Data Types

★Type sizes

◎sizeof (type_name)

- ✧Provides the number of bytes

Using Data Types

- ★Declare all the variables in the beginning of each function
- ★Types need to match!!
  - ◎Declaration
  - ◎Initialization
  - ◎Expressions
  - ◎Placeholders

Flushing the Ouput

- ★printf() sends data to a buffer
- ★Data goes from the buffer to the screen when
  - ◎the buffer is full
  - ◎the output contains a newline
- ★Use fflush() to send the data in the buffer to the screen