UNIVERSITY OF MASSACHUSETTS DARTMOUTH

ECE160: Foundations of Computer Engineering I

Lecture #7 – Formatted Input scanf()

Instructor: Dr. Liudong Xing
SENG-213C, lxing@umassd.edu
ECE Dept

Administrative Issues

- Lab#3
  - Starting on Monday, Feb. 6
- Homework #2 assigned today
  - Due **Friday, Feb. 10**
  - Please follow the "submission guidelines" available in the course website to submit your answers to your name folder at the class M: drive if you haven't
  - Late submission is subject to penalty.

Review of Lecture #6

- Formatted output function printf()
  - Conversion codes %d %c %f etc
  - Field width specification
  - Flag modifiers: 0 and –
  - Use of special characters in printf()
  - Common errors

Topics

- **Formatted input scanf()**

Textbook: chapter 7.4

scanf()

- The C function for reading input from the user is **scanf** (*scan f*ormatted)

scanf(format string, address list)

Example:

```
scanf("%d %f", &student_age, &student_GPA);
```

Note: Microsoft Visual Studio requires using scanf_s()

Format String

- Conversion specifiers / codes

%d	**- integer**
%f	**- float**
%lf	**- double (where lf="long float")**
%c	**- character**

There is no precision width in the input field specification. When scanf() finds a precision, it stops processing.

The Input List

```
scanf("%d %f", &student_age, &student_GPA);
```

- For each conversion code in the format string there must be exactly one address in the address list.
- Each variable name is preceded by **&,** an operator meaning "**the address of**".
- Example: &student_age tells the scanf function to store what it reads from the user at the memory address of student_age.

Do not forget to put &

Rules (1)

- There must be a field specification (conversion specifier) for each field/variable that is going to be read.
- Do not end the format string with a white space character. The program will probably not run.
- With the exception of the character conversion code %c, scanf() skips leading whitespace (leading spaces, tabs, newlines)
- To skip leading white space when reading character data, put a space before the field specification: " %c"

Rules (2)

- The conversion operation processes until
  - End of file is reached <ctrl + z> or <ctrl + d>
  - The maximum number of characters (indicated by the field width, e.g. %3d) have been processed
  - A whitespace character is found after a digit in a numeric specification
  - An error is detected, e.g. a nonnumeric character is found when trying to read a number

Examples

- Data to be input: 100 100.2 1

  scanf(“%d %f %d”, &a, &b, &c);

  - Note: the whitespace between the field specifications are not necessary with numeric input, but it’s good to include them!

- Data to be input: 02/10/91

  scanf(“%2d/%2d/%2d”, &a, &b, &c);

  - Note: the slashes (/) in the format string are not a part of the field specifications, the user must enter them exactly as shown or scanf will stop reading

Exercises (1)

```
int a = 1;
int b = 2;
int c = 3;
scanf(“%d %d”, &a, &b, &c);
printf(“%d %d %d”, a, b, c);
```

If the input is 7 8 9

what is the output?

Choose one of the following

a) 1 2 3

b) 7 8 9

c) 7 8 3

d) 1 2 9

Exercises (2)

What is the output of this program if the input is 100?

```
int c = 0;
scanf(“%d”, c);
printf(“%d”,c);
```

Exercise (3)

- What is the displayed output when the following code fragment is run and the input is the numbers 20 and 30?

```
int       x, y;
printf(“My name is”);
printf(“ Jane Doe.”);
printf(“\nEnter two integers> ”);
scanf(“%d%d”,&x, &y);
x = x + 3;
y = x + y;
printf(“Thanks! The answer is %d.\nBye now!”,y);
```

Exercises (4)

- What is the output of this program if the input is 77.31?

```
float a=2.1;
scanf(“%5.2f”, &a);
printf(“%5.2f”, a);
```

Exercises (5)

- What, if anything, is printed from the following statements, given that x =2 and y =5?

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

Common Programming Errors (1)

- Putting a semicolon after main() is a compilation error
- Forgetting to terminate a comment with */ is a compilation error.
- Forgetting to close the format string in printf or scanf is a compilation error
- Using the incorrect conversion code for the data being read or written is a run-time error.
- Not including required libraries is a linker error.

Common Programming Errors (2)

- Spelling incorrectly the name of functions or reserved words. This produces a compilation error.
- Forgetting the comma after the format string is a compilation error.
- Using commas in the format string of scanf usually results in error.
- Forgetting & in scanf results in error

Good Programming Style

- Adequate white space
- Indentation
- Meaningful variable names
- Comments

Summary of Lecture #7

- Formatted input function scanf()
  - Function format
  - Rules
  - Examples
- Common programming errors

Things To Do

- Review lecture notes and related readings in the textbook
- Homework

Next Topic

- Expressions