CSE 142
Computer Programming I
Input and Output (I/O)

Overview
Topics
Output: printf
Input: scanf
Basic format codes
More on initializing variables

Writing Useful Programs
It’s hard to write useful programs using only variables and assignment statements
Even our Fahrenheit to Celsius program needed more:
Needed a way to get data into and out of the program
We’ll learn more about doing this today
Lots of terminology and messy details, but worthwhile.

What’s a Computer?
This is a computer.

Basic Definitions
Input: movement of data into memory from outside world (e.g., from keyboard).
Changes the value of a variable “read” operation
Output: movement of data from memory to outside world (e.g., to monitor)
“write” operation
Does not change value of memory

Text Output
I/O Statements from a Familiar Program

```c
printf("Enter a Fahrenheit temperature: ");
scanf("%lf", &fahrenheit);
celsius = (fahrenheit - 32.0) * 5.0 / 9.0;
printf("That equals %f degrees Celsius.", celsius);
```

Display Input and Output

The functions `printf` and `scanf` provide display I/O services.

```
printf("control string", list of expressions);
scanf("control string", list of &variables);
```

- **Control string** gives the format of output or input.
- **Expressions** are what to output.
- **Variables** are where to store the input.
- `&` is magic (that is REQUIRED for scanf!)

Display Input and Output

```c
int numPushups;
numPushups = 5;
printf("Hello.  Do %d pushups. 
", numPushups);
output: Hello. Do 5 pushups.
```

What Does the ‘\n’ Do?

```c
int numPushups;
numPushups = 5;
printf("Hello.");
printf("  Do %d pushups. 
", numPushups);
printf("Do them now. 
");
output: Hello. Do 5 pushups. Do them now.
```

Getting a Little Fancier

```c
printf("control string", list of expressions);
```

printf might have more than one expression in its list:

```
printf("%d times %f is %f. 
", multiplier, pi, (double) multiplier * pi);
```

Multiple Output Expressions

% placeholders in format string match expressions in output list in number, order, and type.

```c
int multiplier;
double pi;
pi = 3.14;
multiplier = 2;
printf("%d times %f is %f. 
", multiplier, pi, (double) multiplier * pi);
```

Output: 2 times 3.14000 is 6.28000.
**Advanced Output Formatting**

This is only the beginning! A few of many other things you can do:

- Control number of decimals
  - 3.1 vs 3.100000
- Exponential (scientific) or decimal notation
  - 3.1 vs 3.1E0
- Control total width (including spaces)
  - ___3.1 vs __3.1

How?

Look in textbook or a reference manual, or online help!

---

**Output Format Examples**

<table>
<thead>
<tr>
<th>Format Example</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>%10.2f</td>
<td>___1 2 3 . 5 5 double</td>
</tr>
<tr>
<td>%10.4f</td>
<td>___1 2 3 . 5 5 0 0</td>
</tr>
<tr>
<td>%.2f</td>
<td>1 2 3 . 5 5</td>
</tr>
<tr>
<td>%10d</td>
<td>___4 7 5 int</td>
</tr>
<tr>
<td>%10c</td>
<td>___________ a char</td>
</tr>
</tbody>
</table>

---

**scanf(): Read Input**

```c
scanf ("control string", &input list) ;
int numPushups ;
printf ("Hello.  Do how many pushups? ") ;
scanf (" %d ", &numPushups) ;
printf ("Do %d pushups.
", numPushups) ;
```

output: Hello.  Do how many pushups?  5
Do 5 pushups.

Input list variables MUST be preceded by an &.

---

**Whitespace**

```
space (" "), tab ("\t"), newline ("\n") are “whitespace”
```

Whitespace is skipped by scanf for int ("%d"), and double ("%lf")

This means the user can type spaces before a number and they are ignored

**Not skipped for char input "%c"**

Each character typed, including spaces and newlines, is read separately

---

**Multiple Inputs**

Basic rule:

% placeholders in the format must match variables in the input list

**MUST!** match one-for-one in number, order, and type.

```c
int studentID ;
double grade ;
scanf (" %d %f", &studentID, &grade) ;
```
Input Errors

What happens if the user doesn’t type the right thing for scanf?
- Number with a decimal point when integer expected...
- Character when number expected...

Answer: scanf halts - doesn’t change corresponding variables

Can we detect this when it happens?
Yes, but let’s wait on that.

Format Items Summary

<table>
<thead>
<tr>
<th>Type</th>
<th>scanf()</th>
<th>printf()</th>
</tr>
</thead>
<tbody>
<tr>
<td>char</td>
<td>%c</td>
<td>%c</td>
</tr>
<tr>
<td>int</td>
<td>%d</td>
<td>%d</td>
</tr>
<tr>
<td>double</td>
<td>%lf</td>
<td>%f (long) float</td>
</tr>
</tbody>
</table>

What happens if types don’t match?
- printf → garbled output
- scanf → unpredictable errors

Remember the & !

printf/scanf Summary

Output: printf("control string", output list);
- output list – expressions; values to be printed
- control string – types and desired format
  for now, NO "&", ever!

Input: scanf("control string", &input list);
- input list – variables; values to be read
- control string – types and expected format
  can be a way of initializing variables
  for now, YES "&", always!

Both: %x’s, I/O list match in number, order, type

I/O Summary

Input is the movement of data into memory
In C, we use scanf for input from the keyboard

Output is the movement of data from memory
In C, use printf for output to the screen

Know the basic printf/scanf rules, and know them well

Be aware that advanced formatting options exist
and can be looked up when needed

Review Topic: More on Initializing Variables

Review: Initialization means giving something a value for the first time.

Potential ways to initialize:
- Assignment statement
- scanf

Yet another way: initializer with declaration

Initializing when Declaring

Declarations without initializers
- int product, i;
  product = 40; i = 5;

Declarations with initializers
- int product = 40, i = 5;
  i = 6;

Initializers are part of the declaration;
they are not assignment statements (despite the = sign).
Initialization Quiz

```c
int main (void) {
    /*line 1*/
    int a, b, c, d=10; /*line 2*/
    b=5; /*line 3*/
    d=6; /*line 4*/
    scanf("%d %d", &b, &c); /*line 5*/
    return 0; /*line 6*/
}
```

Q: Where is each of a, b, c, and d initialized?

Next Time

We'll learn about a powerful new type of statement, the conditional or "if" statement.

QOTD: Madlibf

"Madlibs" are games where a person asks for a series of words of different types and then puts those out-of-context words into a story.

Write a scanf statement that would read all the appropriate underlined entries for the following Madlibf:

"I just got my grade back in C S E 142. It was a 3.2! That's a B letter grade! I tell you, I feel 100% better about that class, now!"

Don't get it? Check out http://w3.one.net/~mariae/madlib.htm for an example.