Lecturer-9

C Basics

We’ll Learn Today

- printf() Function
- Format Specifiers
- Escape Sequences
- scanf() Function
- Constants
- Operators

printf() Function

C uses formatted output. The printf function has a special formatting character (%) -- a character following this defines a certain format for a variable:

%c – characters
%d – integers
%f – floats

e.g.

```
printf(``%c %d %f'',ch,i,x);
```

NOTE: Format statement enclosed in ``...'', variables follow after. Make sure order of format and variable data types match up.

Format Specifiers

%c	Single Character
%s	String
%d	Signed decimal integer
%f	Floating point (decimal notation)
%e	Floating point (exponential notation)
%u	Unsigned decimal integer
%x	Unsigned hexadecimal integer
%o	Unsigned octal integer

l prefix used with %d, %u, %x, %o to specify long integer (e.g. %ld)

%Lf	long double

Format Specifiers (con’t)

<u>Example:</u>

```
void main (void)

{

     int event = 5;
     char heat = ‘C’;
     float time = 27.25;
     printf(“ The winning time in heat %c”, heat);
     printf(“ of event %d was %.2f”, event, time);
getch();
}
```

Escape Sequences

The following list shows the common escape sequences:

\n	New line
\t	Tab
\b	Backspace
\r	Carriage return
\’	Single quote
\”	Double quote
\\	Backslash

scanf() Funtion

Syntax:

```
scanf (" format specifier ", & var_name);
```

- Obtains a value from the user
- scanf() uses standard input (usually keyboard)
- This scanf statement has two arguments
  - Format specifier -indicates format of the data
  - & var_name - location in memory to store variable
  - & is confusing in beginning – for now, just remember to include it with the variable name in scanf statement
- When executing the program the user responds to the scanf statement by typing in a number, then pressing the enter (return) key

scanf() Funtion (con’t)

Example:

```
#include<stdio.h>
     main()
{
     int num1,num2,res;
          printf(“Enter First Number: “);
          scanf(“%d”,&num1);

          printf(“Enter Second Number: “);
          scanf(“%d”,&num2);

               res=num1+num2;

          printf(“The sum of two numbers is: %d”, res);
}
```

constant

- ANSI C allows you to declare *constants*. When you declare a constant it is a bit like a variable declaration except the value cannot be changed.
- The *const* keyword is to declare a constant, as shown below:

```
    int const a = 1;
    const int a =2;
```

Note: You can declare the const before or after the type. Choose one an stick to it.

- It is usual to initialise a const with a value as it cannot get a value *any other way*.

Operators

Definition:

“Operators are words or symbols that cause a program to do something to variables.”

There are may different kinds of operators, but most common of them are as follows:

- Arithmetic Operators
- Arithmetic Assignment Operators
- Increment/Decrement Operators
- Relational Operators

Operators (con’t)

Arithmetic:

C operation	Arithmetic operator	Algebraic expression	C expression
Addition	+	$f + 7$	f + 7
Subtraction	-	$p - c$	p - c
Multiplication	*	bm	b * m
Division	/	x / y	x / y
Modulus	%	$r \bmod s$	r % s

Rules of Operator Precendence:

Operator(s)	Operation(s)	Order of evaluation (precedence)
()	Parentheses	Evaluated first. If the parentheses are nested, the expression in the innermost pair is evaluated first. If there are several pairs of parentheses “on the same level” (i.e., not nested), they are evaluated left to right.
*, /, or %	Multiplication,Division, Modulus	Evaluated second. If there are several, they are evaluated left to right.
+ or -	Addition Subtraction	Evaluated last. If there are several, they are evaluated left to right.

Operators (con't)

Arithmetic Assignment:

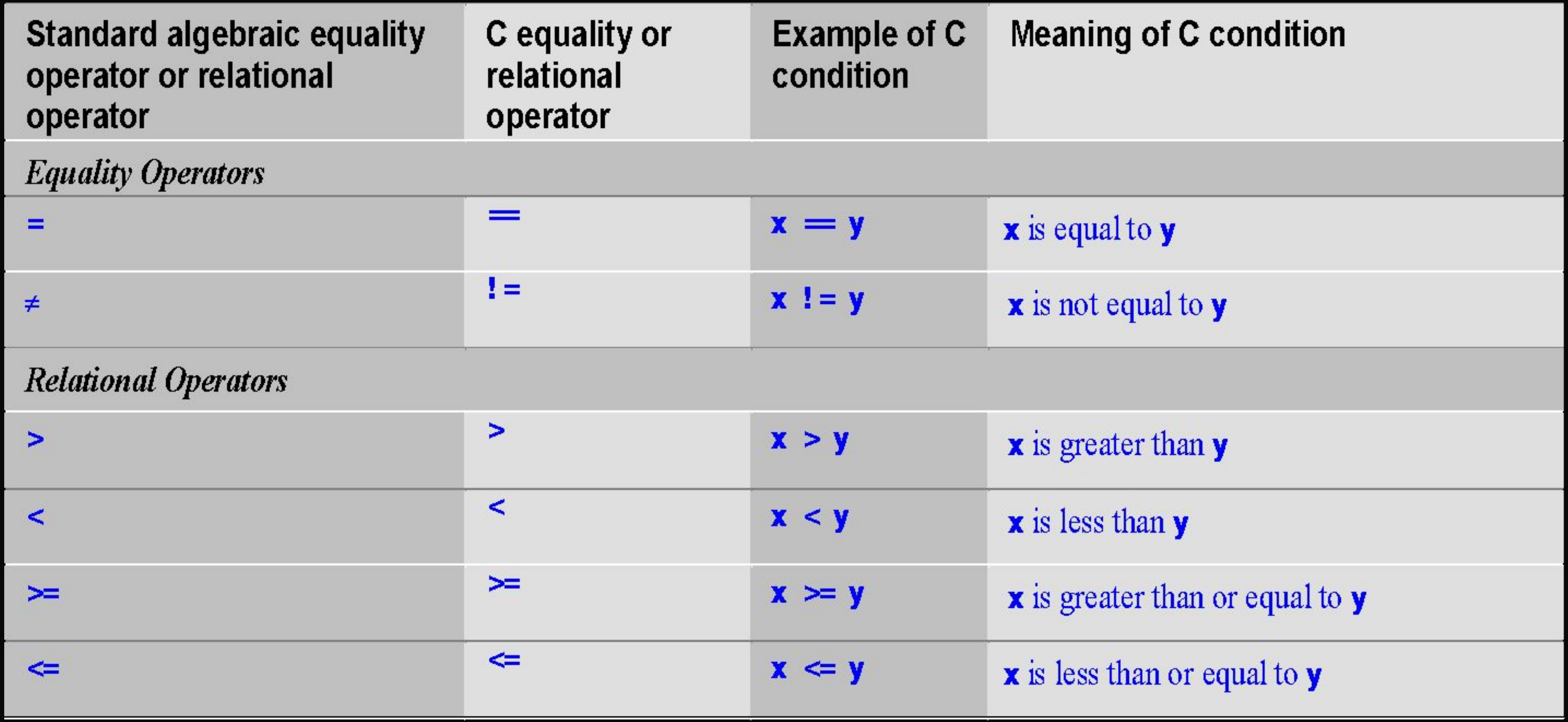
C operation	Arithmetic operator	C expression
Equal	=	a=b
Addition Assignment	+=	a+=b (same as a=a+b)
Subtraction Assingment	- =	a- =b (same as a=a- b)
Multiplication Assignment	*=	a*=b (same as a=a*b)
Division Assignment	/ =	a/ =b (same as a=a/ b)
Remainder Assignment	%=	a%=b (same as a=a%b)

Increment / Decrement:

C operation	Arithmetic operator	C expression
Increment	++	a++ or ++a
Decrement	- -	a- - or - -a

Operators (con’t)

Equality and Relational:



Standard algebraic equality operator or relational operator	C equality or relational operator	Example of C condition	Meaning of C condition
Equality Operators			
=	==	x == y	**x** is equal to **y**
≠	!=	x != y	**x** is not equal to **y**
Relational Operators			
>	>	x > y	**x** is greater than **y**
<	<	x < y	**x** is less than **y**
>=	>=	x >= y	**x** is greater than or equal to **y**
<=	<=	x <= y	**x** is less than or equal to **y**

Assignments

1. Given as input an integer number of seconds, print as output the equivalent time in hours, minutes and seconds. Recommended output format is something like

   7322 seconds is equivalent to 2 hours 2 minutes 2 seconds.

2. Write a program that asks the user to enter two numbers, obtains them from the user and prints their sum, product, difference, quotient and remainder.

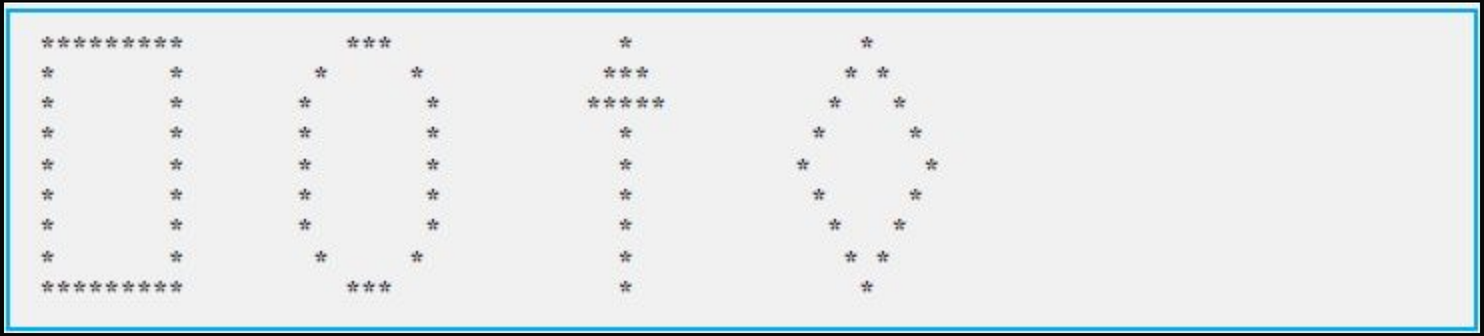
3. Write a program that reads in the radius of a circle and prints the circle's diameter, circumference and area. Use the constant value 3.14159 for π. Perform each of these calculations inside the printf statement(s) and use the conversion specifier %f.
4. Write a program that prints the numbers 1 to 4 on the same line. Write the program using the following methods.

   a) Using one printf statement with no conversion specifiers.

   b) Using one printf statement with four conversion specifiers.

   c) Using four printf statements.

5. Write a program that prints the following shapes with asterisks.



```
*********     ***       *        *
*       *   *     *    ***      * *
*       *  *       *  *****    *   *
*       *  *       *    *     *     *
*       *  *       *    *    *       *
*       *  *       *    *     *     *
*       *  *       *    *      *   *
*       *   *     *     *       * *
*********     ***       *        *
```

6. Create a BMI calculator application that reads the user’s weight in pounds and height in inches, then calculates and displays the user’s body mass index.

$$BMI = \frac{weightInPounds \times 703}{heightInInches \times heightInInches}$$

End of Lecture

Any Questions?