

CS 101 LAB #3

STRUCTURED PROGRAM DEVELOPMENT (PART 1)

Mythili Vutukuru
IIT Bombay



Reference: “C How to Program”, Deitel and Deitel, 8th Edition, Chapter 3 & 4

GOAL OF LAB #3

- In this lab, you will write programs to help you understand the following concepts:
 - Practice with floating point
 - Basic structured programming using if-else, while loop
 - Printing tables and patterns using while loops
- Submit any two programs, testing two different concepts from those mentioned above
- Please write code neatly, with proper indentations and comments.



FLOATING POINT ARITHMETIC

3.19 (*Interest Calculator*) The simple interest on a loan is calculated by the formula

```
interest = principal * rate * days / 365;
```

The preceding formula assumes that `rate` is the annual interest rate, and therefore includes the division by 365 (days). Develop a program that will input `principal`, `rate` and `days` for several loans, and will calculate and display the simple interest for each loan, using the preceding formula. Here is a sample input/output dialog:

```
Enter loan principal (-1 to end): 1000.00
Enter interest rate: .1
Enter term of the loan in days: 365
The interest charge is $100.00

Enter loan principal (-1 to end): 1000.00
Enter interest rate: .08375
Enter term of the loan in days: 224
The interest charge is $51.40

Enter loan principal (-1 to end): -1
```



FLOATING POINT ARITHMETIC

3.18 (*Sales-Commission Calculator*) One large chemical company pays its salespeople on a commission basis. The salespeople receive \$200 per week plus 9% of their gross sales for that week. For example, a salesperson who sells \$5000 worth of chemicals in a week receives \$200 plus 9% of \$5000, or a total of \$650. Develop a program that will input each salesperson's gross sales for last week and will calculate and display that salesperson's earnings. Process one salesperson's figures at a time. Here is a sample input/output dialog:

```
Enter sales in dollars (-1 to end): 5000.00
Salary is: $650.00

Enter sales in dollars (-1 to end): 1234.56
Salary is: $311.11

Enter sales in dollars (-1 to end): -1
```

3.41 (*Diameter, Circumference and Area of a Circle*) Write a program that reads the radius of a circle (as a float value) and computes and prints the diameter, the circumference and the area. Use the value 3.14159 for π .



IF-ELSE, WHILE LOOP

3.22 (*Checking if a Number is Prime*) A prime number is any natural number greater than 1 that is divisible only by 1 and by itself. Write a C program that reads an integer and determines whether it is a prime number or not.

3.23 (*Find the Largest Number*) The process of finding the largest number (i.e., the maximum of a group of numbers) is used frequently in computer applications. For example, a program that determines the winner of a sales contest would input the number of units sold by each salesperson. The salesperson who sells the most units wins the contest. Write a pseudocode program and then a program that inputs a series of 10 non-negative numbers and determines and prints the largest of the numbers. [*Hint:* Your program should use three variables as shown below.]

counter:	A counter to count to 10 (i.e., to keep track of how many numbers have been input and to determine when all 10 numbers have been processed)
number:	The current number input to the program
largest:	The largest number found so far

3.26 (*Find the Two Largest Numbers*) Using an approach similar to Exercise 3.23, find the *two* largest values of the 10 numbers. [*Note:* You may input each number only *once*.]



IF-ELSE, WHILE LOOP

4.9 (*Sum and Average of Integers*) Write a program to sum a sequence of integers and calculate their average. Assume that the first integer read with `scanf` specifies the number of values to be entered. Your program should read only one value each time `scanf` is executed. A typical input sequence might be

7 678 234 315 489 536 456 367

where the 7 indicates that the subsequent 7 values are to be summed.

4.11 (*Calculating the Sum of Multiples*) Write a program to calculate and print the sum of all multiples of 7 from 1 to 100.

4.13 (*Natural Numbers Arithmetic*) Write a program that prints the sum, the sum of the squares, and the sum of the cubes of all natural numbers from 1 till any number entered by the user.

4.14 (*Factorials*) The *factorial* function is used frequently in probability problems. The factorial of a positive integer n (written $n!$ and pronounced “ n factorial”) is equal to the product of the positive integers from 1 to n . Write a program that evaluates the factorials of the integers from 1 to 5. Print the results in tabular format. What difficulty might prevent you from calculating the factorial of 20?



WHILE LOOP, PRINTING TABLES

3.24 (*Tabular Output*) Write a program that uses looping to print the following table of values. Use the tab escape sequence, `\t`, in the `printf` statement to separate the columns with tabs.

N	N^2	N^3	N^4
1	1	1	1
2	4	8	16
3	9	27	81
4	16	64	256
5	25	125	625
6	36	216	1296
7	49	343	2401
8	64	512	4096
9	81	729	6561
10	100	1000	10000



WHILE LOOP, PRINTING PATTERNS

3.32 (*Square of Asterisks*) Write a program that reads in the side of a square and then prints that square out of asterisks. Your program should work for squares of all side sizes between 1 and 20. For example, if your program reads a size of 4, it should print

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

3.33 (*Hollow Square of Asterisks*) Modify the program you wrote in Exercise 3.32 so that it prints a hollow square. For example, if your program reads a size of 5, it should print

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



WHILE LOOP, PRINTING PATTERNS

3.39 (*Checkerboard Pattern of Asterisks*) Write a program that displays the following checkerboard pattern:

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

Your program must use only three output statements, one of each of the following forms:

```
printf( "%s", "* " );
printf( "%s", " " );
puts( "" ); // outputs a newline
```



PLAYING THE NUMBER CLAP GAME

- You may have played the following number clap game as a child. Kids count numbers starting from 1 up to some maximum value N (say 100), taking turns one at a time. That is, the first kid says 1, next one says 2, and so on. However, if the number to be said is a multiple of 7 or has the digit 7 in it, then the kid should simply clap instead of saying the number itself. Write a program that takes N as input from the user, and prints out what is spoken by the kids as they play this game until they reach the number N . Sample output is shown for $N = 20$.
 - Hint: use the mod operator to check for multiples of 7
 - Hint: use the integer division and mod operators to separate out the digits in a number, so that you can see if one of them is 7

```
Enter N: 20
1
2
3
4
5
6
CLAP
8
9
10
11
12
13
CLAP
15
16
CLAP
18
19
20
```

