

# **Object Oriented Programming**

**CSE-208(L)**

**Lab: 01**

**Structured Programming Refresher**

# Agenda for Today

- Welcome to Object Oriented Programming Laboratory!
- Lab Description
  - What material will we cover?
  - What am I getting myself into?
- Administrative Issues
  - Web Page, Text Book, Exams, Office Hours, Homework, Grading, Cheating , Lecture Policy, etc.
- Reading
  - C++ Programming Using Turbo C++ by Robert Lafore
  - How to Program C++ by by Pentice Hall, 2000
  - Herbert Schildt, C++ form the Ground Up, 4TH Edition.
  - (Harvey & Paul) Deitel & Deitel C++ How to Program, 6/E

✓ **Lab participation will help your grade!**

# Lab Discipline

✓ In an effort to make this Lab enjoyable for everybody...

- Please be on time to Lab! This will only benefit YOU!
- Please do not talk to your friends and neighbors in class! It disturbs everyone, and makes it hard to concentrate
- Humming and giggling is unethical in any congregation
- If you have any queries, just ask me or write it down!
- Please turn your pagers and cell-phones off! If you have any important or expected one, attend it outside the Lab

# Software

- For the course, you may use any IDE you are comfortable using. I will use one or more of the following in the Laboratory:
  - Turbo C++
  - Visual C++
  - Dev C++
  - Code Blocks
- All these products can be downloaded from the web for free.
- The class website has instructions on downloading and installing these programs
- If you do not have your own computer, the computer labs on campus have the software

# Structure Programming

Reinstate the concepts of structured programming such as

- **Decision making**
- **Iterations**
- **Recursion**
- **Functions**
- **Structures and Arrays**

✓ You have done these previously, if not that might be problem for you!

# Activity No. 01

- Write a function that reverses elements stored in an array
- Sample OUTPUT of the function

Enter length of the array: 5

Enter the elements of the array:

11      22      33      44      55

Input array is:

11      22      33      44      55

Reverse of the input array is:

55      44      33      22      11

- Note: Actual input array elements are to be swapped i.e. in the above example, swap array element 0 with element 4 and so forth. Temporary array for reversing the elements is not allowed.

## Activity No. 02

- The greatest common divisor of integers  $x$  and  $y$  is the largest integer that evenly divides both  $x$  and  $y$ .
- Write a recursive function `gcd` that returns the greatest common divisor of  $x$  and  $y$ .
- The `gcd` of  $x$  and  $y$  is defined recursively as follows:

If  $y$  is equal to 0, then  $\text{gcd}(x, y)$  is  $x$ ; otherwise  $\text{gcd}(x, y)$  is  $\text{gcd}(y, x \% y)$ , where  $\%$  is the modulus operator.

## Activity No. 03

- A palindrome is a number or a text phrase that reads the same backwards as forwards.
- For example, each of the following five-digit integers is a palindrome: 12321, 55555, 45554, and 11611.
- Write a program that reads in a five-digit integer and determines whether it is a palindrome.

(Hint: use the division and modulus operators to separate the number into its individual digits).



## Activity No. 04

- A company wants to transmit data over the telephone, but they are concerned that their phones may be tapped. All of their data are transmitted as four-digit integers. They have asked you to write a program that encrypts their data so that it can be transmitted more securely. Your program should read a four-digit integer and encrypt it as follows: Replace each digit by (sum of that digit plus 7) modulus 10. Then swap the first digit with the third, swap the second digit with the fourth and print the encrypted integer.
- Write a separate program that inputs an encrypted four-digit integer and decrypts it to form the original number.

## Activity No. 05

- Input an integer containing only 0s and 1s (i.e. a “binary” integer) and prints its decimal equivalent.
- (Hint: use the modulus and division operators to pick off the “binary” number’s digits one at a time from right to left. Just as in the decimal number system where the rightmost digit has a positional value of 1 and the next digit left has a positional value of 10, then 100, then 1000 etc., in the binary number system, the rightmost digit has a positional value of 1, the next digit left has a positional value of 2, then 4, then 8, etc. Thus the decimal number 234 can be interpreted as  $4*1+3*10+2*100$ . The decimal equivalent of binary 1101 is  $1*1 + 0*2+1*4+1*8$  or  $1+0+4+8$  or 13.