CL-1002
Programming Fundamentals
Lab # 12

Objectives:

- Selection sort
- Static variable, local/global variable
- Practicing user defined function
- Pass by value
- Pass by reference
- Function Overloading

Note: Carefully read the following instructions (*Each instruction contains a weightage*)

1. Solve your tasks in same sequence as given in lab manual.
2. First think about statement problems and then write your program.
3. **Write Program in C compiler/IDE and save source file for each program.**
4. Do not copy from any source otherwise your tasks will not accept.
5. Complete your lab **within given Time Slot**. Late submission only accepted with at least 20% deduction (even a single minute late).
   - Submission deadline for CS – 11:30 am
   - Submission deadline for SE – 04:30 pm
6. Add your source code in this **word document (take screen shots of code + output)** + Make one **ZIP file of your all source codes**.
7. Please submit your **Both files** with this naming convention ROLLNO_SECTION_LABNO.
8. Submit your lab on Google Classroom.

Problems

1. Write a C++ program that create two following functions: **Marks (2)**
   1. *void SquarePbyV(int);* //Function 1
   2. *void SquarePbyF(int&);* //Function 2
   You need to get an integer value from user and then pass value as a parameter one by one to both functions to get the square of the value. You need to make 4 function calls as per the given order, and also cout the integer value in main after each function call you make:
   • Function 1
   • Function 2
   • Function 2
   • Function 1
   Sample Input: Enter the number: 2
   Expected Output:
   Square from function1: 4
   Value from main: 2
   Square from function2: 4
   Value from main: 4
   Square from function2: 16
   Value from main: 16
   Square from function1: 256
   Value from main: 16

2. Write a C++ program that will create two same size arrays of Size 10, then you need to fill the both arrays by taking values from user. You need to pass these two arrays to function

   int SimilarCheck(int[ ], int[ ]) that will compare the both elements of arrays with each other's and should return the count of elements which are common in both arrays. **Marks (3)**

   Let's Assume:

   int ArraysA = {1,2,3,4,5,6,6,7,8,9};

   int ArraysB = {1,2,3,4,5,6,6,7,8,10};

   Expected Output:

   Common Elements are: 9

3. You need to implement this problem using functions, you are only allowed to declare and initialize variables/arrays and function call in main. Rest off all logic should be in function including taking input from user and other requirements.

   This program can be used to assign seats for a commercial airplane. The airplane has 13 rows, with six seats in each row. Rows 1 and 2 are first class, rows 3 through 7 are business class, and rows 8 through 13 are economy class. Your program must prompt the user to enter the following information: **Marks (6)**

   1. Ticket type (first class, business class, or economy class)

   2. Desire seat Number

Output the seating plan in the following form:

	A	B	C	D	E	F
Row 1	*	*	X	*	X	X
Row 2	*	X	*	X	*	X
Row 3	*	*	X	X	*	X
Row 4	X	*	X	*	X	X
Row 5	*	X	*	X	*	*
Row 6	*	X	*	*	*	X
Row 7	X	*	*	*	X	X
Row 8	*	X	*	X	X	*
Row 9	X	*	X	X	*	X
Row 10	*	X	*	X	X	X
Row 11	*	*	X	*	X	*
Row 12	*	*	X	X	*	X
Row 13	*	*	*	*	X	*

Here, * indicates that the seat is available; X indicates that the seat is occupied. Make this a menu driven program; show the user's choices and allow the user to make the appropriate choices.

Note: **If seat already occupy show the proper message and then say choice again, and need to do all other exception handlings.**

4. **A person can have the following attributes - name, age, salary and experience. Name and age remain same throughout the code but salary change from organization to organization. Initially experience is zero (0) and after joining organization it can increase by (1).**
   Your task is to create a function name organization, access it multiple times from main() and show attributes after each call in built-in function. (use static, local and global variables)
   Marks (3)

   Example:

   Name- Ali

   Age- 22

   Salary – 99000 (assign any number initially, salary should by increased by 10% in built-in function)

   Experience – 0 (initially, but increase 1 point after each function call)