

**Lab 04**  
**Structure**

## OBJECTIVE:

Things that will be covered in today's lab:

- File Input and Output Streams
- Structures

## THEORY:

### Structs:

We have already discussed array which is a structured data type having all elements of the same type. Another structured data type is a *struct* (or *record*) which allows you to group related values that are of different types.

A *struct* is a collection of fixed number of components, members, in which the members are accessed by name. The members may be different data types.

### Defining a Structure:

A *struct* is a user defined data type that requires the keyword *struct* followed by the name chosen for the *struct*. The data members that define *struct* are contained between a set of curly brackets. *Struct* definition must be ended with a semicolon.

```
struct struct_name
{
    //
    // Member variables
    //
};
```

### Accessing Structure Members:

To access any member of a structure, we use the *member access operator* (.). The member access operator is coded as a period between the structure variable name and the structure member that we wish to access. You would use the *struct* keyword to define variables of structure type.

**Example:** What should be the output of this program?

```
struct uni_t {
char * title ;
int year;
};
int main () {
uni_t uni;

uni.title= "FAST University";
uni.year = 2014;

cout << uni.title;
cout <<" ("<< uni.year <<")\n";
return 0;
}
```

### Exercise 1:

Write a program that reads a telephone number from a file *phnum.txt* (available in lab folder) in the form xxx-xxxx. The first three digits represent an area code and the next 4 digits represent the phone number. Your task is to print these numbers into another file (*outFile.txt*) without any space or '-' between the area code and the number. Your program should define a function that has input and output file streams as arguments.

For example,

If the phone number is 042-5610, the output should be 0425610.

### Exercise 2:

Define a struct, **Item**, with two components: *name* and *price* of the item.

Write a program to help FAST-NUCES restaurant automate its breakfast billing system. The program should do the following:

Show the customer different breakfast items offered by the restaurant. Assume that the restaurant offers the following breakfast items (the price of each item is shown to the right of the item):

- Bacon and Egg           \$2.45
- French Toast           \$1.99
- Fruit Basket           \$2.49
- Cereal                   \$0.69
- Coffee                   \$0.50
- Peanut butter:         \$0.80

## Lab Manual of Programming Fundamentals

Allow the customer to select more than one item from the menu. Calculate and print the bill. Your program must contain at least the following functions:

1. **getData:** This function loads the data into the array *menuList* and number of items may vary.
2. **showMenu:** This function shows the different items offered by the restaurant and tells the user how to select the items.
3. **printCheck:** This function calculates and prints the check.

(Note that the billing amount should include a 5% tax.)The customer can select multiple items of a particular type. A sample output in this case is:

Welcome to FAST Restaurant

1. Bacon and Egg \$2.45
2. Muffin \$1.98
3. Coffee \$0.50

Tax \$0.25

Amount Due \$5.18

### Post Lab:

Write a program that allows two players to play the tic-tac-toe game. Your program must contain the struct **TicTacToe** to implement a *TicTacToe* object. Include a 3-by-3 two-dimensional array, as a member variable, to create the board. If needed, include additional member variables. Some of the operations on a *TicTacToe* object are

- Printing the current board
- Getting a move
- Checking if a move is valid
- Determining the winner after each move.

Add additional operations as needed.