University of Central Punjab

(Incorporated by Ordinance No. XXIV of 2002 promulgated by Government of the Punjab)
FACULTY OF INFORMATION TECHNOLOGY

Programming Fundamentals Section B16
Session: Spring 2025

Assignment no TWO:

😊 **DON'T PANIC** 😊

Part One: Due <u>Handwritten</u> on Tue <u>April 29, 2025</u> in Class

Some Portion of Part One is Due as Homework 4

and

Due in Class on Tue April 22, 2025

Part Two: Portal Submission at 23:59 PM SUN <u>April 27, 2025</u>

The Use of ChatGPT is Allowed for Understanding

Copy (Even One line) from ChatGPT or Anyone or Anywhere Else is Zero Marks

Marks as Many as You Desire

Use of <u>string</u> results in no Grading of Assignment

Use of <u>cstring library (strlen, strcpy etc.)</u> results in no Grading of Assignment

Use of <u>getline and single character reading (get, getc, getch etc.)</u> results in no Grading of Assignment

Final Marks are based on Assignment Evaluation and Formal Quiz Two

Topic	Variables, If-else, loop, 1D Array, File handling and Functions
Objective/ Outcome	• Use of arrays with text files • Converting Standalone Code to Function-Based Code
# of Problems	**Part One (no coding):** 1. **Function Analysis: Set One: 07 Problems; Set Two: 10 Problems ; Set Three: 07 Problems** 2. **Code Dry Run: Five Codes** **Part Two (coding): Set One: 07 Problems; Set Two: 10 Problems ; Set Three: 07 Problems**

General Instructions

- 😊 **DON'T PANIC** 😊
- Assignment may seem long but it contains tasks that you have already done in assignment one
- This assignment is just reorganization of assignment one code in functions

- Make sure to read the **whole assignment document** carefully and take **notes**
- **DON'T DISREGARD INSTRUCTIONS**
- Do as many tasks as you can by the **DEADLINE** as there is no extension in the deadline and **ABSOLUTELY** no Submission will be accepted after the deadline
- Recommended to submit work at least one hour before the deadline to avoid any internet or portal issues
- **CORRECTION** of any **FOUND ERRORS** or **MISTAKES** is **part of the assignment**

Portal Submission Instructions

- Each Set must be in a separate CPP file that should compile and run
- **NO ZIP OR RAR FILES (ZERO MARKS)**
- Name each CPP file for each problem must be as per the following convention:
    - F24_PF_**section**_Registration_A2 _set#.cpp (25 % marks deduction if this is not followed)
    - For example .CPP file for Set 1 of registration L1S24BSCS001 of Section B10 **MUST** have this name:
      F24_PF_B10_ L1S24BSCS001_A2_Set1.cpp
- In each CPP file: (25 % marks deduction if not followed)
    - The first lines in comments must contain: your name, registration, assignment number and problem number
    - AND copy the problem statement
- **Must use the provided following template or ZERO Marks**

Here is a template CPP file for problem one

```
/* 1. File name F24_PF_B10_ L1S24BSCS001_A2_Set1.cpp
Ahmad Usman L1F23BSCS0012
PF Section B10 Assignment Two Set One

Problem One (example only)
Write complete C++ code that keeps reading (positive) prices from a user until user enters a negative
price and then displays sum, average, maximum and minimum price value
The program must produce output as per the following sample output screen:

Enter prices (positive) values or a negative value to stop: 234.24 349.65 789.25 298.12 456.23 98.7
123.34 56.96 -12.3
You have entered 8 prices
Sum is 2406.49 and Average is 300.811
Maximum Price is 789.25 and Minimum Price is 56.96

Problem two
…………………………………..
*/
```

```
#include <iostream>
#include <fstream>
using namespace std;
// 2. Write Function Prototypes here
// 3. Main Function
int main()
{
        // write code here
return 0;
}

// 4. Codes of all Functions
```

Marking Scheme:

- Each Day Assignment is Late is 25 % marks deduction per day
- Each Problem marks are Listed with the Problem:
    o Correct Logic: 30 % marks
    o Correct Code: 20 % marks
    o Correct Syntax: 20 % Marks
    o Correct Output: 30 % marks
    o Not Obeying any instruction in **red is ZERO Marks**
- Marks Deductions:
    o Not Following Instructions (up to 100 % marks deductions)
    o Not Following Good Code Writing Practices as Listed below (2 marks per violation)
        ▪ Meaningful names of variables
        ▪ Each variable in a separate line
        ▪ Each variable must be initialized
        ▪ Use of Indentation and White spaces to make code readable
            Each if, else, while, for, do-while etc. must have brackets { }
- Handwritten on Time Submission 60 %
- **Use of Advance C++ than Covered in Class is zero Marks**
- **Copy is ZERO for All Parties**
- **Copy from ChatGPT or anywhere elase iz ZERO marks**
- Code does not **compile or run**: **ZERO MARKS**

PART ONE: Handwritten only

(Don't Use any Code or Diagrams, Explain Everything in Pure English)

1. **Function Analysis:** You will need to read the task two of assignment One and do a function analysis of each set. You will need to write the name of functions to use and for each function: parameters and data types, passed by value or reference, return type. **So, for each set, consider**

at least as many functions as many tasks in the set, for example, set one must have at least 7 functions. Must use the format provided in the following example [5 Marks per Function]

<u>EXAMPLE</u>: Write a complete program that reads integer values from user for 2 arrays and then inserts the first array in the second array

<u>EXAMPLE SOLUTION</u>:

Function Names: read, readArray, shiftRightOnce, shiftRight, Insert, display

Function Details:

read: reads a single integer value from the user

    Parameters: None

    Return type: int

readArray: reads N integers values from the user

    Parameters:

        Parameter 1: int array to hold integer values

        Parameter 2: int (by value) for size of array

    Return type: void

shiftRightOnce: shifts Once an array right from a given index

    Parameters:

        Parameter 1: int array to hold integer values

        Parameter 2: int (by value) for size of array

        Parameter 3: int (by value) for index

    Return type: void

shiftRight: shifts an array right from a given index and by a given amount

    Parameters:

        Parameter 1: int array to hold integer values

        Parameter 2: int (by value) for size of array

        Parameter 3: int (by value) for index

        Parameter 4: int (by value) for amount

    Return type: void

insert: Inserts one array in another array

>   **Parameters:**
>
>>   **Parameter 1: int source array**
>>
>>   **Parameter 2: int (by value) for size of source array**
>>
>>   **Parameter 3: int destination array**
>>
>>   **Parameter 4: int (by value) for size of destination array**
>
>   **Return type: void**

display: display an integer array on screen

>   **Parameters:**
>
>>   **Parameter 1: int array**
>>
>>   **Parameter 2: int (by value) for size of array**
>
>   **Return type: void**

2. For each of the following **Five Codes**, do a dry Run (**must use the same format and method as provided in the document previously shared on portal). [20 Marks per Code]**

   # Identify any errors and correct these error

Code 1:

```
1.   int count(int values[], int size, int v)
2.   {
3.     int cnt = 0;
4.     for(int i = 0 ; i < size; i++)
5.      {
6.        if(values[i] == v)
7.             {
8.                cnt++;
9.             }
10.   }
11.   return cnt;
12. }
13.
14. int main()
15. {
16.   int set[20] = {3, 1, 2, 2, 3, 3, 1, 2, 3, 4, 3, 3};
17.   int n = 12;
18.   int freq = count(set, n, set[5]);
19.   cout << "Frequency of " << set[5] << " is " << freq << endl;
```

```
20.   return 0;
21. }
```

Code 2

```
1.    void read(char s[])
2.    {
3.      cout << "Enter an English word: ";
4.      cin >> s;
5.    }
6.
7.    void length(char s[], int & len)
8.    {
9.      len = 0;
10.   while(s[len++] != '\0');
11. }
12.
13. void copyStr(char src[], char dst[])
14. {
15.   int ls = 0;
16.   length(src, ls);
17.
18.   for(int i = 0; i < ls; i++)
19.     {
20.       dst[i] = src[i];
21.     }
22. }
23.
24. bool isLowercase(char c)
25. {
26.   bool result = false;
27.   result = (c >= 'a');
28.   result = result && (c <= 'z');
29.   return result;
30. }
31.
32. bool isUPPERCASE(char c)
33. {
34.   bool result = false;
35.   result = (c >= 'A');
36.   result = result && (c <= 'Z');
37.   return result;
38. }
```

![University of Central Punjab logo]

University of Central Punjab
(Incorporated by Ordinance No. XXIV of 2002 promulgated by Government of the Punjab)
FACULTY OF INFORMATION TECHNOLOGY

```
39.
40. void changeCase(char s[])
41. {
42.   int ls = 0;
43.   length(s, ls);
44.   for(int i = 0; i < ls; i++)
45.    {
46.      if(isLowercase(s[i]))
47.          {
48.            s[i] -= 32;
49.          }
50.      if(isUPPERCASE(s[i]))
51.          {
52.            s[i] += 32;
53.          }
54.    }
55. }
56.
57. int main()
58. {
59.   char word[30] = "";
60.   read(word);
61.   char save[30] = "";
62.   copyStr(word, save);
63.   changeCase(word);
64.   cout << save << " " << word << endl;
65.
66.   return 0;
67. }
```

Code 3

```
1.   void length(char s[], int & len)
2.   {
3.     len = 0;
4.     while(s[len++] != '\0');
5.   }
6.
7.   void append(char cstr[], char word[])
8.   {
9.     int ls = 0;
10.    length(cstr, ls);
11.    int lw = 0;
12.    length(word, lw);
13.    int i = 0;
14.    int si = ls - 1;
```

```
15.   int ei = si + lw;
16.   for(i = si; i < ei; i++)
17.   {
18.      cstr[i] = word[i - si];
19.   }
20. }
21.
22. void readPath(char path[])
23. {
24.   cout << "Enter file path: ";
25.   cin >> path;
26. }
27. void readPara(char path[], char para[], int & wordCount)
28. {
29.   char word[30] = "";
30.   char space[] = " ";
31.   ifstream R(path);
32.   while(R >> word)
33.   {
34.      append(para, word);
35.      append(para, space);
36.      wordCount++;
37.   }
38.   R.close();
39. }
40.
41. int main()
42. {
43.   char file[80] = "";
44.   readPath(file);
45.   char para[200] = "";
46.   int wc = 0; // word count
47.   readPara(file, para, wc);
48.   cout << para << endl;
49.   return 0;
50. }
```

Code 4:

```
1.   void isFound(int values[], int size, int v, bool & b)
2.   {
3.     b = false;
4.     for(int i = 0 ; i < size; i++)
5.     {
6.        if(values[i] == v)
7.             {
```

```
8.              b = true;
9.              return;
10.          }
11.     }
12. }
13.
14. void removeDuplicates(int values[], int size, int single[], int & cnt)
15. {
16.   cnt = 0;
17.   for(int i = 0; i < size; i++)
18.    {
19.      int v = values[i];
20.      bool B = false;
21.      isFound(single, cnt, v, B);
22.      if(!B)
23.          {
24.            single[cnt] = v;
25.            cnt++;
26.          }
27.    }
28. }
29.
30. void readPath(char path[])
31. {
32.   cout << "Enter file path: ";
33.   cin >> path;
34. }
35.
36. void readFile(char path[], int values[], int & no)
37. {
38.   no = 0;
39.   ifstream R(path);
40.   while(R >> values[no++]);
41.   R.close();
42. }
43.
44. void display(int values[], int size)
45. {
46.   for(int i = 0; i < size; i++)
47.    {
48.      cout << values[i] << " ";
49.    }
50.   cout << endl;
51. }
52.
```

```
53. int main()
54. {
55.   char file[80] = "";
56.   readPath(file);
57.   int prices[100] = {0};
58.   int noOfPrices = 0;
59.   readFile(file, prices, noOfPrices);
60.   display(prices, noOfPrices);
61.   int set[20] = {0};
62.   int count = 0;
63.   removeDuplicates(prices, noOfPrices, set, count);
64.   display(set, count);
65.   return 0;
66. }
```

Code 5

```
1.   void count(char s[], char ch, int & cnt)
2.   {
3.     cnt = 0;
4.     for(int i = 0 ; s[i] != '\0'; i++)
5.     {
6.       if(s[i] == ch)
7.             {
8.               cnt++;
9.             }
10.    }
11. }
12.
13. void frequencies(char str[], int freq[])
14. {
15.   for(int i = 0; str[i] != '\0'; i++)
16.   {
17.     char c = str[i];
18.     count(str, c, freq[i]);
19.   }
20. }
21.
22. void display(char chars[], int freq[])
23. {
24.   cout << "Characters Frequencies\n";
25.   for(int i = 0; chars[i] != '\0'; i++)
26.   {
27.     cout << "  " << chars[i] << "\t\t" << freq[i] << endl;
28.   }
29.   cout << endl;
```

```
30. }
31.
32. int main()
33. {
34.   char cstr[30] = "abcdeabcefabrt";
35.   int freq[30] = {0};
36.   frequencies(cstr, freq);
37.   display(cstr, freq);
38.   return 0;
39. }
```

PART TWO: CODING[Each Function of each Set is of 5 marks and Total marks are based on number of Functions done]:

In this part you will **use your code of assignment one part two** and **reorganize the code in functions**. If you missed any code in assignment one then here is a chance to do those now.

In this assignment two part two, you will write **functions from each task of each Set of assignment one**, with the following constraints:

- Main must not contain any cin, cout, file reading or writing and mathematics
- Must Use at least **one function for each of the task** of a set. For example, set one must have at least 7 individual functions. **Since each function is of 5 marks, using more functions will result in more marks.**
- Must follow the format of the provided template for each of the code file

☺Don't Panic ☺
Good Luck