**COMSATS University Islamabad**

**Lahore Campus**

****

DEPARTMENT OF COMPUTER SCIENCE

Programming Fundamentals

LAB TASK (CSC103)

PREPARED BY

Abdul Karim Shahid,

Assistant Professor

Create a DOC file to upload your work on teams

SP23-BCS-A:

Use following information to join class at teams (<https://teams.microsoft.com/l/team/19%3auGUZDZ9PRbKls2RR6m-tU5T46LKQTuRLb54FLdYMBAQ1%40thread.tacv2/conversations?groupId=127bc2b7-16e5-4c19-86b5-c0ab80851141&tenantId=75df096c-8b72-48e4-9b91-cbf79d87ee3a>).

SP23-BCS-B:

Use following information to join class at teams (<https://teams.microsoft.com/l/team/19%3aZkGnlMiKgJQNLA0-dJmXudkmVgT9qXG4AmVg4-rCcHc1%40thread.tacv2/conversations?groupId=0e8e5331-e52a-4bb7-8c1c-73f91093727b&tenantId=75df096c-8b72-48e4-9b91-cbf79d87ee3a>)

**Doc File Format for labwork submission**

**Title Page:**

**Lab Work XX**

CSC103-Programming Fundamentals



Submitted by:

**NAME**

**SP23-BCS-XXX**

**A/B**

Submitted to:

**Mr. Abdul Karim Shahid**

Submitted on: **March 3, 2023**

**Department of Computer Science**

**COMSATS University Islamabad**

**Lahore Campus**

**Lab task Solution:**

* Problem statement
* Flowchart or Pseudo code (if asked)
* Solution (C code must contain Your name and registration number)
* Screen shot of sample output with white background

**CSC103 Programming Fundamentals Lab**

***Week1&2: The Programming Environment***

**Learning Objectives:**

The objectives of this first experiment are:

1. To make you familiar with the Windows programming environment and the "C" compiler. For this purpose you will use DevCPP/CodeBlocks/Turbo C (TC). You will write, compile and run a (it may be your first) programme in "C"

**Your lab report is expected to contain the following for each exercise:**

* **C Source Code with proper comments.**
* **At the top of your C source code also write the following comments**
  + **/\* This programme is prepared by XXXXXXXXXXX Reg no.: SP23-BCS-XXX Section: A/B on dd/mm/yyyy. This programme ……….(a bit of explanation what your programme does.\*/**
* **Formatted Output and the printf function**

One of the common task in every program is the printing of output. We use the output to request input from a user and later display the status/result, computations etc. In C programming there are several functions for printing formatted output. Here we discuss the printf() function, which writes output to the computer monitor. To use the printf() function we must include the stdio.h library in the source code. To do this just place the following code at the beginning of your program.

#include<stdio.h>

To print a simple message in computer screen you might call printf() function as follows:

* #include <stdio.h>
* int main()
* {
* printf ("You are learning printf() function");
* return 0;
* }

Output:

You are learning printf() function

In the above examples, the cursor will remain at the end of the printed output. If you repeat the above code in the following way the second message would appear immediately after the first one.

* #include <stdio.h>
* int main()
* {
* printf("You are learning printf() function");
* printf("You are learning printf() function");
* return 0;
* }

Output:

You are learning printf() function You are learning printf() function

If we want to print the second output in a new line we must need a different way, which has been discussed in the following section.

* **Escape sequences in C**

An escape sequence is a series of characters that represents a special character. It begins with a backslash character (\), which indicates that the character(s) that follow the backslash character should be treated in a special way. C uses escape sequences within a format string to print certain special characters. For example \n moves the output position to the beginning of the next line. The following is a list of escape sequences.

|  |  |
| --- | --- |
| **Escape sequence** | **Action** |
| \n | prints a new line |
| \b | backs up one character |
| \t | moves the output position to the next tab stop |
| \\ | prints a backslash |
| \" | prints a double quote |
| \' | prints a single quote |

You will get an idea of using the above escape sequences from the following example.

* #include<stdio.h>
* int main()
* {
* printf("Create a new line\n");
* printf("Print a double quotes (\") within a string\n");
* printf("Print a single quotes (\') within a string\n");
* printf("Print a Backslash\\ within a string\n");
* printf("Using Backspace\b within a string\n");
* printf("Using\tTab within a string\n");
* return 0;
* }

Output:

* Create a new line
* Print a double quotes (") within a string
* Print a single quotes (') within a string
* Print a Backslash\ within a string
* Using Backspace within a string
* Using     Tab within a string

**Exercise 1:** Print following shape using simple printf statements (You may print these shapes vertically in one program)

(1) (2) (3) (4) (5) (6)

\* \*\*\*\*\*\*\*\*\*\* \* \* \*\*\*\*\* \*\*\*\*\*  
 \*\*\* \* \* \*\* \*\* \*\*\*\* \*\*\*\*  
 \*\*\*\*\* \* \* \*\*\* \*\*\* \*\*\* \*\*\*  
 \*\*\* \* \* \*\*\*\* \*\*\*\* \*\* \*\*  
 \* \*\*\*\*\*\*\*\*\*\* \*\*\*\*\* \*\*\*\*\* \* \*

**Exercise 2:** Write a program that prints the numbers 1 to 4 on the same line. Write the program using the following methods.

1. Using one printf statement with no conversion specifiers.
2. Using one printf statement with four conversion specifiers. printf("%d%d%d%e",1,2,3,4) //Here %d is the conversion specifier use to print integer values on the screen
3. Using four printf statements

**Exercise 3:** Write a C-Program to perform the simple arithmetic operations (addition, subtraction, multiplication, division, remainder).

**Exercise 4:** Write a C-Program to swap two integer numbers without and with using third variable.

**Exercise 5:** Write a C-Program to calculate area and Perimeter of the triangle.

[Area of triangle= ½ x base x vertical height]

[Perimeter of triangle = a + b + c]

Declare variable base, vh and total\_area

Calculate area of triangle total\_area = 0.5 \* base \* vh

Print total\_area of triangle triangle