


National University of Computer and Emerging Sciences



Programming Fundamentals CS118 Laboratory Manual

Course Instructor	Mirza Mubasher Baig
Lab Instructor#01	Fraz Yousaf
Lab Instructor#02	Samia Akhter
Section	BDS-1A
Semester	FALL 2021

National University of Computer and Emerging Sciences

	Lab No 8		
	Course Name	Programming Fundamentals	Course Code CS188
	Program	BS(DS)	Semester Fall 2021
	Duration	2.5 hours	Total Points
	Lab Date	12-Nov-2021	Weight 3%
	Section	BDS-1A	Page(s) 11

Topics Covered: Functions

Use Visual Studio to write code for the following questions.

Submission Guidelines:

1. Save all .cpp files according to the following naming convention i.e.,
{Section}_{RollNo}_{ProblemNo}.cpp
For Problem#01: CX_21L-XXXX_P01.cpp For Problem#02: CX_21L-XXXX_P02.cpp
.
.
For Problem#06: CX_21L-XXXX_P06.cpp
2. Now create a new folder according to the following naming convention i.e.,
{Section}_{ROLLNO}_{LABNO}
For students of C1: C1_21L-XXXX_L07 For students of C2: C2_21L-XXXX_L07
3. Move all of your .cpp files to this newly created directory and compress it into a single .zip file.
4. Submit this compressed file on Google Classroom.

Even one-minute late submission would be considered as late and won't be accepted

Functions in C++

C++ Function Declaration

The syntax to declare a function is:

```
returnType functionName (parameter1, parameter2,...) {  
    // function body  
}
```

Here's an example of a function declaration.

```
// function declaration  
void greet() {  
    cout << "Hello World";  
}
```

Here,

- the name of the function is `greet()`
- the return type of the function is `void`
- the empty parentheses mean it doesn't have any parameters
- the function body is written inside `{}`

Note: We will learn about `returnType` and `parameters` later in this tutorial.

Calling a Function

In the above program, we have declared a function named `greet()` . To use the `greet ()` function, we need to call it.

Here's how we can call the above `greet()` function.



How Function works in C++

(Don't submit code for this problem)

Practice Problem

Write a function named *isArmstrong* that takes an integer parameter *number* and returns boolean **true** if the number is Armstrong, otherwise **false**.

An Armstrong number is the sum of its own digits each raised to the power of total number of digits. For example

153 is an example of 3-digit Armstrong number because:

$$153 = 1^3 + 5^3 + 3^3 = (1*1*1) + (5*5*5) + (3*3*3) = 1 + 125 + 27 = 153$$

In above example, we raised the digits to the power of 3 because total no. of digits in 153 is 3

1634 is an example of 4-digit Armstrong number because:

$$1634 = 1^4 + 6^4 + 3^4 + 4^4 = (1*1*1*1) + (6*6*6*6) + (3*3*3*3) + (4*4*4*4) = 1 + 1296 + 81 + 256 = 1634$$

In above example, we raised the digits to the power of 4 because total no. of digits in 1634 is 4

Submit codes for questions starting from here

*****Enjoy Coding ☐*****

Problem#01

Marks (10)

Write a function named **isOdd** that takes an integer parameter **number** and returns string **TRUE** if the number is even, otherwise **FALSE**.

Sample Input#1:

5

Sample Output#1:

TRUE

Sample Input#2:

-4

Sample Output#2:

FALSE

Sample Input#3:

16

Sample Output#3:

FALSE

Sample Input#4:

21

Sample Output#4:

TRUE

Skeleton Code:

```
#include<iostream>
using namespace std;

string isOdd(int number) {
    //Write your Code here
}

int main() {
    int n;
    cin >> n;
    cout << isOdd(n)<< endl;
    return 0;
}
```

Problem#02

Marks (20)

Copy this code to your cpp file and implement the functions.
These functions do not return any value(**void**);

```
int main()
{
```

```
#include<iostream>
using namespace std;

int main() {

    int a = 10, b = -17;

        Addition(a, b);
        Multiplication(a, b);
        Table(a);
        Square(a);
return 0;
}
```

```
}
```

The sum of 10 and -17 is -7
The multiplication of 10 and -17 is -170
Display Table 1 to 10
Square of a= 100

Problem#03

Marks (20)

Write a function named ***digitMul*** which takes a positive integer ***num*** as a parameter of the function. The function is responsible to return the Multiply of its individual digits.

For Example:

Digit sum of 123 = $1*2*3 = 6$

Digit sum of 5672 = $5*6*7*2 = 420$

Sample Input#1:

156

Sample Output#1:

30

Sample Input#2:

10091

Sample Output#2:

0

Sample Input#3:

1

Sample Output#3:

1

Sample Input#4:

99

Sample Output#4:

81

Problem#04

Marks (20)

Write a function named ***i_am_joking*** ☐ which takes a positive integer ***num*** as a parameter of the function. The function is responsible to keep taking the individual digit sum until it's not possible any further (the sum has reached to a single digit). Your function should return the last one digit obtained at the end and display it in the main function.

For Example:

If user enters 1279 as an input, the output should be 1

Step#1	1279	=>	1 + 2 + 7 + 9	=>	19
Step#2	19	=>	1 + 9	=>	10
Step#3	10	=>	1 + 0	=>	1

If user enters 999 as an input, the output should be 9

Step#1	999	=>	9 + 9 + 9	=>	27
Step#2	27	=>	2 + 7	=>	9

Sample Input#1:

9219

Sample Output#1:

3

Sample Input#2:

572

Sample Output#2:

5

Sample Input#3:

99994

Sample Output#3:

4

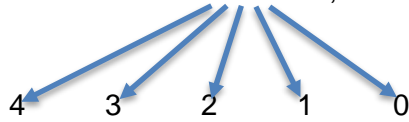
Problem#05

Marks (20)

Write a function named ***findNthDigit*** which takes two integers ***number*** and ***index*** as the function parameters and then returns the Nth Digit of that number.

For Example:

Let's assume we have a number 97895, then its indexing would be like



If the value for index given to user is 3, the function should return 7.

Sample Input#1:

5678

0

Sample Output#1:

8

Sample Input#2:

5678

2

Sample Output#2:

6

Sample Input#3:

10

Sample Output#3:

0

Problem#06

Marks (10)

Write a function named **swap** which takes two integers **alpha** and **beta** as the function parameters and swap their values. The swapping must be visible outside the **swap** function too. After the calling of swap function, display values of the two variables.

Input Format:

alpha in first row
beta in second
row

Output Format:

alpha in first row
beta in second
row

Sample Input#1:

52
22

Sample Output#1:

22
52

Sample Input#2:

-1
9

Sample Output#2:

9
-1

Sample Input#3:

4
2

Sample Output#3:

2
4