**National University of Computer and Emerging Sciences, Lahore Campus**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Course:** | **PF Lab** | **Course Code:** | **CL118** |
|  | **Program:** | **BS (Computer Science)** | **Semester:** | **Fall 2018** |
|  | **Duration:** | **30 Minutes** | **Total Marks:** | **15** |
|  | **Paper Date:** | **11-Oct-18** | **Weight** | **4 %** |
|  | **Section:** | **C** | **Page(s):** | **1** |
|  | **Exam:** | **Quiz 1** | **Reg. No.** |  |

**Instruction/Notes:** Honesty always gives fruit and Dishonesty is always harmful.

# Quesrtion#1

a. What is the difference between passing an argument by value and by reference in a function.

b. What is Function Overloading.

Pass by Value: The method parameter values are copied to another variable and then the copied object is passed, that's why it's called pass by value.

Pass by Reference: An alias or reference to the actual parameter is passed to the method, that's why it's called pass by reference.

Function Overloading in C++ You can have multiple definitions for the same function name in the same scope. The definition of the function must differ from each other by the types and/or the number of arguments in the argument list. You cannot overload function declarations that differ only by **return type**.

# Question#2

**Write a C++ that takes a number and detect whether it is Armstrong number or not. Your code should work for any number. No need to write main only function implementation is required.**

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

For example, 153 is a 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

1634 is an example of 4-digit Armstrong number.

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

**Solution**:

int main() {

int num = 153, digitSum, temp, remainderNum, digitNum ;

temp = num;

digitNum = 0;

while (temp != 0) {

digitNum++;

temp = temp/10;

}

temp = num;

digitSum = 0;

while (temp != 0) {

remainderNum = temp%10;

digitSum = digitSum + pow(remainderNum, digitNum);

temp = temp/10;

}

if (num == digitSum)

cout<<num<<" is an Armstrong number";

else

cout<<num<<" is not an Armstrong number";

return 0;

}

## Department of Computer Science Page 1