**Tutorial 2**

**(Class, Object, Constructors and Destructors in C++)**

**Instructions**

1. All tutorial sheets will be posted on the Google Classroom.

2. Students are advised to submit tutorial sheets solutions in classroom.

Q1. Consider a C++ class named Fraction. An object of type Fraction will represent a fraction (a rational number), such as 3/7, -17/5, or 8/1. Given two integers **x**and **y**and where **x** is divisible by **y**. It can be represented in the form of a fraction **x/y**. Your task is to write the definition of the class Fraction and reduce the fraction to its lowest form.

Input : x = 16, y = 10

Output : x = 8, y = 5

Define all member functions fully within the class Fraction. Write a default constructor, parameterized constructor, destructor for the above program.

Q2. Modify the above program to add two fractions using class and objects. The program must comprise of the following functions:

1. To return the greatest common divisor
2. To convert the obtained fraction to its reduced (simplest) form.
3. To add two fractions. Arithmetically, addition of fractions works like this:

*a/b + c/d = (ad+bc) / bd*

Your method must apply reduce() to the sum computed by this formula, before returning.

1. To display fractions taking one argument in a form like 8/3 or -6/7 or 7/1.

Define all member functions fully within the class Fraction. Write a default constructor, parameterized constructor, destructor for the above program.

Q3. Predict the output of the following program (based on constructor order invocation)

#include<iostream>

using namespace std;

class Test

{

int a;

public:

Test() { cout << "Default Constructor called\n"; }

Test(int x) { a=x; cout<<"parameterized Constructor called\n";}

Test(const Test &t) { cout << "Copy constructor called \n"; }

void f1(Test arg1) { cout << "inside f1() \n";}

Test f2(Test &arg2) { cout<<"inside f2() \n"; Test temp1; return temp1; }

};

int main()

{

Test \*t1, \*t2;

t1 = new Test();

t2 = new Test(\*t1);

Test t3 = \*t1;

Test t4;

t4 = t3;

Test t5(7);

Test t6(t5);

t1->f1(t5);

t6=t1->f2(t5);

return 0;

}