**24-March-2021**

**Lab Practice Exercises**

1. Write a C++ program to overload the addition, subtraction, multiplication, division and modulo operations for two complex numbers.
2. Write a C++ program to create a class for fraction using numerator and denominator as its data members. Overload suitable operators to add and subtract two fractions with same denominators and different denominators.
3. Write a C++ program to negate a Distance class having feet and inches using unary operator overloading. Also overload suitable operators to increase the Distance by 10 times.
4. Create a Stack with appropriate push and pop operations. Design the stack in such a way any kind of data should get stored in the stack. (Use Class Templates)
5. Write a C++ program to calculate the area of various polygons using Template overloading.
6. Write a C++ program to generate Fibonacci series using function templates.
7. Illustrate the usage of static variables in class templates and function templates by taking an example of your own.

**Additional Exercises**

**Fill in the missing code**

**Question 1:**

#include<iostream>

#include<cstring>

using namespace std;

class String

{

private:

    char \*s;

    int size;

public:

    String(const char \*str = NULL); // constructor

    ~String() { ------------;  }// destructor

    String(const String&); // copy constructor

    void print() { cout << s << endl; } // Function to print string

    void change(const char \*);  // Function to change

};

String::String(const char \*str)

{

    size = strlen(str);

    s = new char[size+1];

    strcpy(s, str);

}

void String::change(const char \*str)

{

}

String::String(const String& old\_str)

{

}

int main()

{

    String str1("GeeksQuiz");

    String str2 = str1;

    str1.print(); // what is printed ?

    str2.print();

    str2.change("GeeksforGeeks");

    str1.print(); // what is printed now ?

    str2.print();

    return 0;

}

**Question 2:**

#include<iostream>

using namespace std;

class Test

{

private:

  int x;

  int y;

public:

  Test (int x = 0, int y = 0) { }

  Test setX(int a) { x = a; -----------; }

  Test setY(int b) { y = b; -----------------; }

  void print() { cout << "x = " << x << " y = " << y << endl; }

};

int main()

{

  Test obj1;

  obj1.setX(10).setY(20);

  obj1.print();

  return 0;

}

**Predict the** **output and give justifications.**

**Question 3:**

#include<iostream>

using namespace std;

class Test

{

private:

  int x;

  int y;

public:

  Test(int x = 0, int y = 0) { this->x = x; this->y = y; }

  static void fun1() { cout << "Inside fun1()"; }

  static void fun2() { cout << "Inside fun2()"; this->fun1(); }

};

int main()

{

  Test obj;

  obj.fun2();

  return 0;

}

**Question 4:**

using namespace std;

class Test {

    int value;

public:

    Test(int v = 0) {value = v;}

    int getValue() const { return ++value;}

};

int main() {

    Test t(20);

    cout<<t.getValue();

    return 0;

}

**Question 5:**

class Test {

   static Test \* fun()

{

     return this;

   }

};

int main()

{

   getchar();

   return 0;

}

**Question 6:**

#include<iostream>

using namespace std;

class Test

{

private:

    static int count;

public:

    Test& fun();

};

int Test::count = 0;

Test& Test::fun()

{

    Test::count++;

    cout << Test::count << " ";

    return \*this;

}

int main()

{

    Test t;

    t.fun().fun().fun().fun();

    return 0;

}

**Question 7:**

#include<iostream>

using namespace std;

class Point {

public:

    Point() { cout << "Normal Constructor calledn"; }

    Point(const Point &t) { cout << "Copy constructor calledn"; }

};

int main()

{

   Point \*t1, \*t2;

   t1 = new Point();

   t2 = new Point(\*t1);

   Point t3 = \*t1;

   Point t4;

   t4 = t3;

   return 0;

}

**Question 8:**

**A program to demonstrate the concept of constructors and destructor**

#include<iostream.h>

#include<conio.h>

#include<stdlib.h>

class DEPOSIT

{

long int principal;

int time;

float rate;

float totalamount;

public:

DEPOSIT(); // #1

DEPOSIT(long p, int t, float r); // #2

DEPOSIT(long p, int t); // #3

DEPOSIT(long p, float r); // #4

DEPOSIT(const Deposit &d); // #5

~DEPOSIT();

void calculateamount(void);

void display(void);

};

**Question 9:**

**A program to demonstrate the concept of returning objects from a function**

#include<iostream.h>

class weight

{

int kilogram;

int gram;

public:

void getdata ();

void putdata ();

void sum\_weight (weight,weight) ;  
weight sum\_weight (weight) ;

} ;