TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING

PULCHOWK CAMPUS

LAB REPORT ON

OBJECT ORIENTED PROGRAMMING [CT 451]

LAB 2

CLASS AND OBJECT CONCEPT , CONSTRUCTORS & DESTRUCTORS

Submitted by:

Rujal Acharya

PUL076BEI029

Submitted to:

Department of Electronics and Computer Engineering, Pulchowk Campus

Institute of Engineering, Tribhuvan University

Lalitpur, Nepal

November, 2020

Task 1

Problem:

Write a program in CPP to input the name, roll, marks and address of a student from the user and display the entered details using the concept of class and object.

Program:

```
#include <iostream>

using namespace std;

class Student {
	char name[20];
	int roll;
	float marks;
	char address[10];

	public:
	void getdata() {
		cout << "Enter the name, roll, marks and address of the student " << endl;
		cin >> name >> roll >> marks >> address;
	}

	void showdata() {
		cout << ".....Displaying details of the student....." << endl;
		cout << "Name : " << name << endl;
		cout << "Roll : " << roll << endl;
		cout << "Marks : " << marks << endl;
		cout << "Address : " << address << endl;
	}
};

int main() {
	Student student;
	student.getdata();
	student.showdata();
	return 0;
}
```

Task 2

Problem:

Write a program in CPP to input the name, roll, marks and address of n students from the user and display the entered details using the concept of class and objects.

Program:

```
#include <iostream>

using namespace std;

class Student {
	char name[20];
	int roll;
	float marks;
	char address[10];

	public:
	void getdata() {
		cout << "Enter the name, roll, marks and address of the student " << endl;
		cin >> name >> roll >> marks >> address;
	}

	void showdata() {
		cout << "Name : " << name << endl;
		cout << "Roll : " << roll << endl;
		cout << "Marks : " << marks << endl;
		cout << "Address : " << address << endl;
	}
};

class Students {
	Student s[20];
	int n;

	public:
	void getdata() {
		cout << "Enter the number of students : " ;
		cin >> n;
		for (int i = 0; i < n; i++) {
			cout << "For student " << i+1 << endl;
			s[i].getdata();
```

```
            }
        }

        void showdata() {
            for (int i = 0; i < n; i++) {
                cout << ".....Displaying details of student " << i+1 << "....." << endl;
                s[i].showdata();
            }
        }
};

int main() {
        Students students;
        students.getdata();
        students.showdata();
        return 0;
}
```

Task 3

Problem:

Write a program in CPP to find the sum of two complex numbers using the OOP concept.

Program:

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

class Complex {
	float re;
	float imz;

	public:
	void getdata() {
		cout << "Enter the real and imaginary parts" << endl;
		cin >> re >> imz;
	}

	void showdata() {
		if (imz < 0) {
			cout << re << imz << "i" << endl;
		} else {
			cout << re << "+" << imz << "i" << endl;
		}
	}

	void add(Complex a, Complex b) {
		re = a.re + b.re;
		imz = a.imz + b.imz;
	}
};

int main() {
	Complex a,b,sum;
	a.getdata();
	b.getdata();
	sum.add(a, b);
	sum.showdata();
	return 0;
}
```

Task 4

Problem:

Write a program in CPP to illustrate the concept of constructor(default, parameterized and copy constructor) and destructor taking class complex as an example.

Program:

```
#include <iostream>

using namespace std;

class Complex {
	float re;
	float imz;

	public:
	// Default constructor
	Complex() {
		re = 0;
		imz = 0;
		cout << "Inside default constructor with real value " << re << " and imaginary value " << imz << endl;

	}

	// Parameterized constructor
	Complex(float r, float i) {
		re = r;
		imz = i;
		cout << "Inside parameterized constructor with real value " << re << " and imaginary value " << imz << endl;
	}

	// Copy constructor
	Complex(Complex &c) {
		re = c.re;
		imz = c.imz;
		cout << "Inside copy constructor with real value " << re << " and imaginary value " << imz << endl;
	}
```

```
    // Destructor
    ~Complex() {
        cout << "Destroying the object with real part " << re << " and imaginary part " <<
imz << endl;
    }

    void showdata() {
        if (imz < 0) {
            cout << re << imz << "i" << endl;
        } else {
            cout << re << "+" << imz << "i" << endl;
        }
    }

};

int main() {
    Complex a, b(2,-1.5), c(b);
    a.showdata();
    b.showdata();
    c.showdata();
    return 0;
}
```