TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING

PULCHOWK CAMPUS

LAB REPORT ON

OBJECT ORIENTED PROGRAMMING [CT 451]

LAB 5

TEMPLATES AND EXCEPTION HANDLING FEATURES OF C++

Submitted by:

Rujal Acharya

PUL076BEI029

Submitted to:

Department of Electronics and Computer Engineering, Pulchowk Campus

Institute of Engineering, Tribhuvan University

Lalitpur, Nepal

December, 2020

Task 1

Problem:

Write a program in CPP to find the sum of two integer numbers and two float numbers using the concept of function template.

Program:

```
#include <iostream>

template <class T>
T sum (T a, T b) {
	return a + b;
}

int main ( ) {
	int a, b;
	float fa, fb;

	std::cout << "Enter two integers: ";
	std::cin >> a >> b;
	std::cout << "The sum of " << a << " and    " << b << " is " << sum (a, b) << std::endl;

	std::cout << "Enter two floating point numbers: ";
	std::cin >> fa >> fb;
	std::cout << "The sum of " << fa << " and " << fb << " is " << sum (fa, fb) << std::endl;
	return EXIT_SUCCESS;
}
```

Task 2

Problem:

Write a program in CPP to find the area of rectangle with integer and float dimensions using the concept of class template.

Program:

```
#include <iostream>

template <class T>
class Rectangle {
	public:
		Rectangle(T a, T b);
		T area ( );

	private:
		T length;
		T breadth;
};

template <class T>
Rectangle <T> :: Rectangle (T a, T b) {
	this->length = a;
	this->breadth = b;
}

template <class T>
T Rectangle <T> :: area ( ) {
	return this->length * this->breadth;
}

int main ( ) {
	int a, b;
	float fa, fb;

	std::cout << "Enter the length and breadth as integers: ";
	std::cin >> a >> b;
	Rectangle <int> intRect (a, b);
	std::cout << "The area of rectangle is " << intRect.area ( ) << std::endl;

	std::cout << "Enter two floating point numbers: ";
	std::cin >> fa >> fb;
	Rectangle <float> floatRect (fa, fb);
	std::cout << "The area of rectangle is " << floatRect.area ( ) << std::endl;
	return EXIT_SUCCESS;
}
```

Task 3

Problem:

Write a program in CPP to sort the list of n strings in alphabetical order using the concept of STL.

Program:

```
#include <iostream>
#include <list>

int main ( ) {
	int n;
	std::string temp;
	std::list <std::string> strings;

	std::cout << "Enter the number of strings: " ;
	std::cin >> n;

	for (int i = 0; i < n; i++) {
		std::cin >> temp;
		strings.push_back (temp);
	}

	strings.sort ( );

	while (!strings.empty ( )) {
		std::cout << strings.front ( ) << std::endl;
		strings.pop_front ( );
	}

	return EXIT_SUCCESS;
}
```

Task 4

Problem:

Write a program in CPP to handle divide by zero exception using the concept of exception handling.

Program:

```
#include <iostream>

int main ( ) {
	float a, b;
	std::cout << "Enter two numbers: " ;
	std::cin >> a >> b;

	try {
		if (b == 0) {
			throw 0;
		}
		std::cout << a << " / " << b << " = " << a / b << std::endl;
	} catch (int n) {
		if (n == 0) {
			std::cerr << "Divide by zero exception caught..." << std::endl;
			return EXIT_FAILURE;
		}
	}

	return EXIT_SUCCESS;
}
```

Task 5

Problem:

Write a program in CPP to illustrate the concept of rethrowing an exception

Program:

```
#include <iostream>

void divide (float a, float b) {
	try {
		if (b == 0) {
			throw 0;
		}
		std::cout << a << " / " << b << " = " << a / b << std::endl;
	} catch (int n) {
		if (n == 0) {
			std::cerr << "Can't catch the exception, rethrowing...." << std::endl;
			throw;
		}
	}
}

int main ( ) {
	float a, b;
	try {
		std::cout << "Enter the numbers you want to divide: " ;
		std::cin >> a >> b;
		divide (a, b);
	} catch (int n) {
		if (n == 0) {
			std::cerr << "Divide by zero exception caught, quitting..." << std::endl;
			return EXIT_FAILURE;
		}
	}
	return EXIT_SUCCESS;
}
```