CS2006 C++ Lab Exercises
Week 18

1. Write templates for the two functions minimum and maximum. The minimum function should accept two arguments and return the value of the argument that is the lesser of the two. The maximum function should accept two arguments and return the value of the argument that is the greater of the two. Design a simple driver program that demonstrates the templates with various data types.

2. Write a function template that accepts an argument and returns its absolute value. The absolute value of a number is its value with no sign. For example, the absolute value of -5 is 5, and the absolute value of 2 is 2. Test the template in a simple driver program.

3. Write a class named TestScores. The class constructor should accept an array of test scores as its argument. The class should have a member function that returns the average of the test scores. If any test score in the array is negative or greater than 100, the class should throw an exception. Demonstrate the class in a program.

4. A function that returns a special error code is usually better accomplished throwing an exception instead. The following class maintains an account balance.

```
class Account
{
 private:
      double balance;
 public:
      Account()
      {
            balance = 0;
      }
      Account(double initialDeposit)
      {
            balance = initialDeposit;
      }
      double getBalance()
      {
            return balance;
      }
      // returns new balance or -1 if error
      double deposit(double amount)
      {
            if (amount > 0)
                  balance += amount;
            else
                  return -1;        // Code indicating error
            return balance;
      }
      // returns new balance or -1 if invalid amount
      double withdraw(double amount)
      {
            if ((amount > balance) || (amount < 0))
                  return -1;
            else
                  balance -= amount;
            return balance;
      }
};
```

Rewrite the class so that it throws appropriate exceptions instead of returning -1 as an error code. Write test code that attempts to withdraw and deposit invalid amounts and catches the exceptions that are thrown.

5. Modify the SimpleVector class template discussed in the lecture to include the member function push_back. The function push_back should emulate the STL vector class push_back.

6. Write a class named PhoneBookEntry that has members for a person’s name and phone number. Then write a program that creates atleast five PhoneBookEntry objects and stores them in a vector. Use a loop to print all objects in the vector.