EEE102 C++ Programming and Software Engineering II

Lab Practice 5

Functions

Notice:

- The aim of this lab is for you to become familiar with the usage of functions.
- Practice with the exercises. These parts are not for submission.

1. Function Overloading

In C++ two different functions can have the same name if their parameter types or number are different. That means that you can give the same name to more than one function if they have either a different number of parameters or different types in their parameters.

Exercise 1.1 Can the functions below work? If they cannot work properly, explain why.	
1	`bool swap(int &a, int &b);` `int swap(int c, int d);`

2	`int func(int a, int b, int c);`
	`void func(int a, int b);`
	`float func(float a, int b);`
	`float func(int *a, int b);`

3	`class student`
	`{`
	`  string name;`
	`  double avgMark;`
	`public:`
	`  void student();`
	`  void student(string name, double avgMark);`
	`};`

4	`class student`
	`{`
	`  string name;`
	`  double avgMark;`
	`public:`
	`  student();`
	`  student(string ename="N/A", double eMark=0);`
	`};`

```
5   class student
    {
        string name;
        double avgMark;
    public:
        student();
        student(string ename, double eMark);
        ~student(bool bIsEmpty);
    };
```

Exercise 1.2

The class of complex number introduced in lecture 4 is defined below.

```
1   class complexClass
2   {
3       double x;
4       double y;
5   public:
6       ......
7       void operator =(complexClass a);
8       complexClass operator +(complexClass a);
9       complexClass operator -(complexClass a);
10      ......
11  };
```

Part 1: Overload the operators '+''-''*' and '/' as for complex multiplication and division;

Part 2: Overload the operator '^' as for complex power.

2. Default argument

Exercise 2.1

Can the functions below work? If they cannot work properly, explain why.

```
1   // declaration
    int fun1(int a, int b, int c=0);
    // definition
    int fun1(int a, int b, int c=0)
    { . . . }
```

```
2   void fun2(double a, double b=1);
    void fun2(double c);
```

```
3   // declaration
    int fun3(int a, int b, int c=1);
    // called by
    int result1 = fun3(1,2,3);
    int result2 = fun3(5);
    double result3 = fun3(20,50);
```

Exercise 2.2
The effect of a default argument can be alternatively achieved by function overloading.
Design an example to verify.

3. Recursion

A recursive function is a function that calls itself, either directly or indirectly (through another function). One of the most famous applications of recursion is the Fibonacci series:

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, ...

begins with 0 and 1, and has the property that each subsequent Fibonacci number is the sum of the previous two Fibonacci numbers. Therefore, the Fibonacci series can be defined recursively as follows:

Fibonacci(0)=0;

Fibonacci(1)=1;

Fibonacci(n)=Fibonacci(n-1)+Fibonacci(n-2);

Exercise 3
Write a function to calculate the n^{th} Fibonacci number recursively. Use this function to calculate the following Fibonacci numbers: 1,2,3,4,5,10,20,30,40.