ENGG1340 Computer Programming II
COMP2113 Programming Technologies
Module 5 Self-Review Exercise

1. Evaluate the following expressions:
Give the (i) function header; (ii) function prototype (without parameter names), for each of the following functions:
   (a) Function `hypotenuse` that takes two double-precision, floating-point arguments, `side1` and `side2`, and returns a double-precision, floating-point result.
   (b) Function `smallest` that takes three integers, `x`, `y` and `z`, and returns an integer.
   (c) Function `instructions` that does not receive any arguments and does not return a value. [Note: Such functions are commonly used to display instructions to a user.]
   (d) Function `intToDouble` that takes an integer argument, `number`, and returns a double-precision, floating-point result.

Example solution for (a):

(i) `double hypotenuse( double side1, double side2 )`
(ii) `double hypotenuse( double, double );`

(b) (i) `int smallest( int x, int y, int z )`
(ii) `int smallest( int, int, int );`
(c) (i) `void instructions()`
(ii) `void instructions();`
(d) (i) `double intToDouble( int number )`
(ii) `double intToDouble( int );`

2. Define a function `hypotenuse` that calculates the length of the hypotenuse of a right triangle when the other two sides are given. Use this function in a program to determine the length of the hypotenuse for each of the following triangles. The function should take two arguments of type `double` and return the hypotenuse as a `double`.

```
#include <iostream>
#include <cmath>

using namespace std;

double hypotenuse( double, double );

int main()
{
	double side1, side2;

	for ( int i = 1; i <= 3; ++i )
	{
		cout << "\nEnter 2 sides of right triangle: ";
		cin >> side1 >> side2;

		cout << "Hypotenuse: " << hypotenuse( side1, side2 ) << endl;
	}

	return 0;
}

double hypotenuse( double s1, double s2 )
```

```
{
    return sqrt( s1 * s1 + s2 * s2 );
}
```

3. Find the error(s) in each of the following program segments, and explain how the error(s) can be corrected:

```
(a)   int g()
      {
           cout << "Inside function g" << endl;
           int h()
           {
                cout << "Inside function h" << endl;
           }
      }

(b)   int sum( int x, int y )
      {
           int result;
           result = x + y;
      }

(c)   double square( double number )
      {
           double number;
           return number * number;
      }
```

(a) *Error*: Function `h` is defined in function `g`.
Correcion: Move the definition of `h` out of the definition of `g`.

(b) *Error*: The function is supposed to return an integer, but does not.
Correction: Delete variable `result` and place the following statement in the function: `return x + y;`

(c) *Error:* Variable `number` is declared twice.
Correction: Remove the declaration within the `{}`.

4. What is the output of the following program?

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

void find(int a, int &b, int &c);

int main()
{
    int one, two, three;

    one = 5;
    two = 10;
    three = 15;

    find(one, two, three);
    cout << one << ", " << two << "," << three << endl;

    find(two, one, three);
    cout << one << ", " << two << "," << three << endl;
    find(three, two, one);
    cout << one << ", " << two << "," << three << endl;

    find(two, three, one);
    cout << one << ", " << two << "," << three << endl;

    return 0;
}

void find(int a, int& b, int& c)
{
    int temp;

    c = a + b;
    temp = a;
    a = b;
    b = 2 * temp;
}
```

```
5, 10, 15
20, 10, 15
25, 30, 15
45, 30, 60
```

5. Consider the following program that will generate a random number between 1 and 3.

```
int computerChoice = rand() % 3 + 1;
```

Write a program that allows a user to play the Rock Paper Scissors game with computer continuously. Take a look at the following sample run.

```
What do you choose? [1: Rock| 2: Paper| 3: Scissors| 4: Exit]: 1
Computer: 2, User: 1
The computer won!
What do you choose? [1: Rock| 2: Paper| 3: Scissors| 4: Exit]: 2
Computer: 1, User: 2
You won!
What do you choose? [1: Rock| 2: Paper| 3: Scissors| 4: Exit]: 3
Computer: 3, User: 3
It was a tie!
What do you choose? [1: Rock| 2: Paper| 3: Scissors| 4: Exit]: 4
```

```
#include <cstdlib>
#include <iostream>

using namespace std;

int main() {
    // Definition
    // 1: Rock
    // 2: Paper
    // 3: Scissors

    std::cout << "What do you choose? [1: Rock| 2: Paper| 3: Scissors| 4: Exit]: ";
    int userChoice;
    std::cin >> userChoice;
    while (userChoice != 4) {

        int computerChoice = rand() % 3 + 1;

        std::cout << "Computer: " << computerChoice << ", User: " << userChoice << endl;

        if (computerChoice == userChoice) {
            std::cout << "It was a tie!" << endl;
        }

        else if (computerChoice == 1 && userChoice == 3) {
            std::cout << "The computer won!" << endl;
        }

        else if (computerChoice == 2 && userChoice == 1) {
            std::cout << "The computer won!" << endl;
        }

        else if (computerChoice == 3 && userChoice == 2) {
            std::cout << "The computer won!" << endl;
        }

        else {
            std::cout << "You won!" << endl;
        }

        std::cout << "What do you choose?. [1: Rock| 2: Paper| 3: Scissors| 4: Exit] ";
        std::cin >> userChoice;
    }

    return 0;
}
```