

# **CSC 2000**

## **Introduction to C++ Programming Language**

### **Assignment 04**

**50 points**

**Due 03/16/2020 (11:45 A.M.)**

#### Assignment Objectives

#### Assignment Objectives

- . Learn about user-defined functions
- . Examine value-returning functions, including actual and formal parameters
- . Explore how to construct and use a value-returning, user-defined function in a program
- . Examine value-returning functions, including actual and formal parameters
- . Learn how to construct and use void functions in a program
- . Discover the difference between value and reference parameters
- . Explore reference parameters and value-returning functions
- . Learn about the scope of an identifier
- . Examine the differences between local and global identifiers
- . Explore functions with default parameters
- . Learn about function overloading

Answer questions 1 to 6 on a word file; write a program for Q.7 and Q.8.

All assignments must be submitted by the Canvas. **No email or hard copy** is accepted. You must follow the following format:

- a. For non-programming questions, use a word file to type your answers. Don't use the text box on the Canvas to answer the questions or to write comments, we will not read it.
- b. State your answer clearly.
- c. For programming questions, include only the source file for each problem.
- d. Submit your file to the Canvas. You must submit your assignment on time; otherwise, you will receive zero. In addition, you cannot submit your file more than one time.
- e. There will be several folders on the Canvas. You need to upload your file(s) using the correct folder on the Canvas.
- f. Name each file: "Assignment Number(Question number(s))".
- g. To upload your file(s):
  - In Course Navigation, click the Assignments link.
  - Click the title of the assignment.
  - Click the **Submit** Assignment button.
  - Add **File**. ...
  - Add Another **File**. ...
  - **Submit** Assignment. ...
  - View **Submission**.

**It is your responsibility to make sure that each file is uploaded correctly. If you uploaded a wrong file, you receive zero; files will not be accepted after due date even if you have a prove that the file is created before the due date.**

**Make sure you review the Cheating & Plagiarism policy on Canvas.**

1. (10 points)

Identify the following items in the programming code shown below:

- a. Function prototype, function heading, function body, and function definitions.
- b. Function call statements, formal parameters, and actual parameters.
- c. Value parameters and reference parameters.
- d. Local variables and global variables.
- e. Named constants.

```
#include <iostream> //Line 1

using namespace std; //Line 2

const double NUM = 3.5; //Line 3

int temp; //Line 4

void func(int, double&, char); //Line 5

int main() //Line 6
{ //Line 7
    int num; //Line 8
    double one; //Line 9
    char ch; //Line 10

    func(num, one, ch); //Line 11
    cout << num << " " << one << " " << ch << endl; //Line 12
    func(16, one, '%'); //Line 13
    cout << num << " " << one << " " << ch << endl; //Line 14

    return 0; //Line 15
} //Line 16

void func(int first, double& second, char ch) //Line 17
{ //Line 18
    int num; //Line 19
    double y; //Line 20
    int u; //Line 21

    num = 2 * first; //Line 22
    y = second * first; //Line 23
    u = static_cast<int>(ch); //Line 24
    second = num + y * u; //Line 25
} //Line 26
```

2. (4 points)

Consider the following functions:

```
int find(int num)
{
    int first, second;

    first = num * num;
    second = first + num;

    if (second > 100)
        num = first / 10;
    else
        num = first / 20;

    return num + 2;
}

int discover(int one, int two)
{
    int secret = 0;

    for (int i = one; i < two; i++)
        secret = secret + i * i;

    return secret;
}
```

What is the output of each of the following program segments?

- a. `cout << find(15) << endl;`
- b. `cout << discover(3, 9) << endl;`
- c. `cout << find(10) << " " << discover(10, find(10)) << endl;`
- d. `x = 12; y = 8;`  
`cout << discover(y, x) << endl;`

3. (4 points)

Consider the following program:

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

using namespace std;

void traceMe(double x, double y);

int main()
{
    double one, two;

    cout << "Enter two numbers: ";
    cin >> one >> two;
    cout << endl;
    traceMe(one, two);
    traceMe(two, one);

    return 0;
}
```

```

void traceMe(double x, double y)
{
    double z;

    if (x != 0)
        z = sqrt(y) / x;
    else
    {
        cout << "Enter a nonzero number: ";
        cin >> x;
        cout << endl;
        z = floor(pow(y, x));
    }

    cout << fixed << showpoint << setprecision(2);
    cout << x << ", " << y << ", " << z << endl;
}

```

determine the scope of each identifier.

4. (6 points)

Consider the following program. What is its exact output? Show the values of the variables after each line executes.

```

#include <iostream>

using namespace std;

void funOne(int& a);

int main()
{
    int num1, num2;

    num1 = 10; //Line 1

    num2 = 20; //Line 2

    cout << "Line 3: In main: num1 = " << num1
         << ", num2 = " << num2 << endl; //Line 3

    funOne(num1); //Line 4
    cout << "Line 5: In main after funOne: num1 = "
         << num1 << ", num2 = " << num2 << endl; //Line 5

    return 0; //Line 6
}

```

```

void funOne(int& a)
{
    int x = 12;
    int z;

    z = a + x;                                     //Line 7

    cout << "Line 8: In funOne: a = " << a
          << ", x = " << x
          << ", and z = " << z << endl;           //Line 8

    x = x + 5;                                     //Line 9

    cout << "Line 10: In funOne: a = " << a
          << ", x = " << x
          << ", and z = " << z << endl;           //Line 10

    a = a + 8;                                     //Line 11

    cout << "Line 12: In funOne: a = " << a
          << ", x = " << x
          << ", and z = " << z << endl;           //Line 12
}

```

5. (4 points)

Consider the following function definition:

```

void defaultParam(int num1, int num2 = 7, double z = 2.5)
{
    int num3;

    num1 = num1 + static_cast<int>(z);
    z = num2 + num1 * z;
    num3 = num2 - num1;
    cout << "num3 = " << num3 << endl;
}

```

What is the output of the following function calls?

- a. `defaultParam(7);`
- b. `defaultParam(8, 2);`
- c. `defaultParam(0, 1, 7.5);`
- d. `defaultParam(1, 2, 3.0);`

6. (2 points)

Locate as many errors as you can.

```
int getValue()
{
    int inputValue;
    cout << "Enter an integer: ";
    cin >> inputValue;
    return inputValue;
}
double getValue()
{
    double inputValue;
    cout << "Enter a floating-point number: ";
    cin >> inputValue;
    return inputValue;
}
```

## Programming Questions

7. (10 points)

The following formula gives the distance between two points,  $(x_1, y_1)$  and  $(x_2, y_2)$  in the Cartesian plane:

$$\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Given the center and a point on the circle, you can use this formula to find the radius of the circle. Write a program that prompts the user to enter the center and a point on the circle. The program should then output the circle's radius, diameter, circumference, and area. Your program must have at least the following functions:

- a. **distance**: This function takes as its parameters four numbers that represent two points in the plane and returns the distance between them.
- b. **radius**: This function takes as its parameters four numbers that represent the center and a point on the circle, calls the function `distance` to find the radius of the circle, and returns the circle's radius.
- c. **circumference**: This function takes as its parameter a number that represents the radius of the circle and returns the circle's circumference. (If  $r$  is the radius, the circumference is  $2\pi r$ .)
- d. **area**: This function takes as its parameter a number that represents the radius of the circle and returns the circle's area. (If  $r$  is the radius, the area is  $\pi r^2$ .)

Assume that  $\pi = 3.1416$ .

8. (10 points)

Write a program that outputs inflation rates for two successive years and whether the inflation is increasing or decreasing. Ask the user to input the current price of an item and its price one year and two years ago.

To calculate the inflation rate for a year, subtract the price of the item for that year from the price of the item one year ago and then divide the result by the price a year ago.

Your program must contain at least the following functions: a function to get the input, a function to calculate the results, and a function to output the results.

Use appropriate parameters to pass the information in and out of the function. Do not use any global variables.