

## User-Defined Functions II

### Objectives of the Lecture

- Function Prototype.
- Programming Examples.
- Value-Returning Functions: Special Cases.

### Function Prototype

- **Function prototype:** function heading without the body of the function.
- **Syntax:**

```
functionType functionName(parameter list);
```

- It is **not necessary** to specify the variable name in the parameter list.
- The data type of each parameter must be specified.

### Programming Examples

```
// Example 1: find the largest number of a set of 10 numbers.
//*****
#include <iostream>
using namespace std;
double larger(double x, double y);
int main()
{
    double num; //variable to hold the current number
    double max; //variable to hold the larger number
    int count; //loop control variable
    cout << "Enter 10 numbers." << endl;
    cin >> num; //Step 1
    max = num; //Step 1
    for (count = 1; count < 10; count++) //Step 2
    {
        cin >> num; //Step 2a
        max = larger(max, num); //Step 2b
    }
    cout << "The largest number is " << max << endl; //Step 3
    return 0;
} //end main
double larger(double x, double y)
{
    if (x >= y)
        return x;
    else
        return y;
}
```

```

// Example 2: Largest of three numbers
#include <iostream>
using namespace std;
double larger(double x, double y);
double compareThree(double x, double y, double z);
int main()
{
    double one, two; //Line 1
    cout << "Line 2: The larger of 5 and 10 is "
         << larger(5, 10) << endl; //Line 2
    cout << "Line 3: Enter two numbers: "; //Line 3
    cin >> one >> two; //Line 4
    cout << endl; //Line 5
    cout << "Line 6: The larger of " << one
         << " and " << two << " is "
         << larger(one, two) << endl; //Line 6
    cout << "Line 7: The largest of 43.48, 34.00, "
         << "and 12.65 is "
         << compareThree(43.48, 34.00, 12.65)
         << endl; //Line 7
    return 0; }
double larger(double x, double y)
{    if (x >= y)
        return x;
    else
        return y; }
double compareThree (double x, double y, double z)
{    return larger(x, larger(y, z)); }

// Example 3: CourseGrade Function

```

### EXAMPLE 6-3

In this example, we write the definition of function `courseGrade`. This function takes as a parameter an `int` value specifying the score for a course and returns the grade, a value of type `char`, for the course. (We assume that the test score is a value between 0 and 100 inclusive.)

```

char courseGrade(int score)
{
    switch (score / 10)
    {
        case 0:
        case 1:
        case 2:
        case 3:

```

```

    case 4:
    case 5:
        return 'F';
    case 6:
        return 'D';
    case 7:
        return 'C';
    case 8:
        return 'B';
    case 9:
    case 10:
        return 'A';
    }
}

```

You can also write an equivalent definition of the function `courseGrade` that uses an `if...else` structure to determine the course grade.

## Value-Returning Functions: Special Cases

### a) Case 1

```

int secret(int x)
{
    if (x > 5)           //Line 1
        return 2 * x;   //Line 2
}

```

A correct definition of the function `secret` is:

```

int secret(int x)
{
    if (x > 5)           //Line 1
        return 2 * x;   //Line 2

    return x;           //Line 3
}

```

```

#include <iostream>
using namespace std;
int secret(int x);
int main()
{
    int num = 5;
    int a;
    a = 2 + secret(num);
    cout << a << " " << secret(a) << endl; return 0; }
int secret(int x)

```

```

{   if (x > 5)           //Line 1
    return 2 * x;       //Line 2
    return x;           //Line 3
}

```

#### b) Case 2

```

// This program illustrates that a value-returning function
// returns only one value, even if the return statement
// contains more than one expression.

```

```

#include <iostream>
using namespace std;
int funcRet1();
int funcRet2(int z);
int main()
{   int num = 4;
    cout << "Line 1: The value returned by funcRet1: "
         << funcRet1() << endl;           // Line 1
    cout << "Line 2: The value returned by funcRet2: "
         << funcRet2(num) << endl;       // Line 2
    return 0; }
int funcRet1()
{   int x = 45;
    return 23, x; //only the value of x is returned
}
int funcRet2(int z)
{   int a = 2;
    int b = 3;
    return 2 * a + b, z + b; //only the value of z + b is returned
}

```

```

return x, y; //only the value of y will be returned

```

```

int funcRet1()
{
    int x = 45;

    return 23, x; //only the value of x is returned
}

int funcRet2(int z)
{
    int a = 2;
    int b = 3;

    return 2 * a + b, z + b; //only the value of z + b is returned
}

```