Programming Language (630203)
Fall 2010/2011 – Lecture Notes # 17

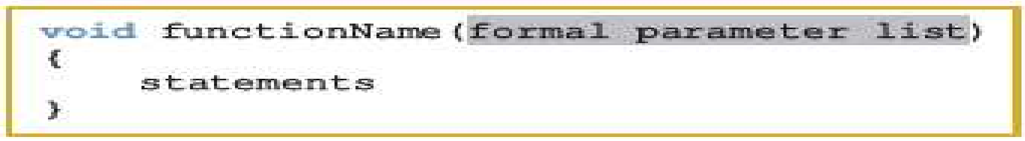
void Functions

Objectives of the Lecture

- **Construct and use void functions.**
- **Menus Design using void functions (Programming Example 1)**
- **Programming Example 2**

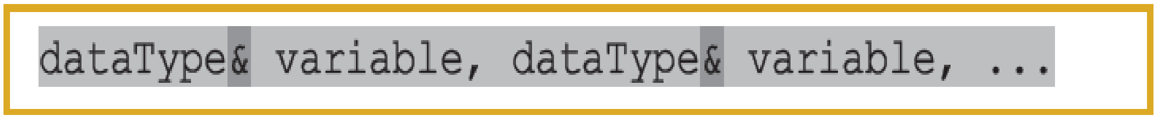
Construct and use void functions

- **void functions** and **value-returning functions** have similar structures.
  - Both have a heading part and a statement part
- User-defined void functions can be placed either before or after the function main
- If user-defined void functions are placed after the function main
  - The function prototype must be placed before the function main.
- A void function does not have a **return type**
- **return statement without any value** is typically used to exit the function early
- Formal parameters are **optional**
- A call to a void function is a **stand-alone statement**
- **Function definition syntax:**



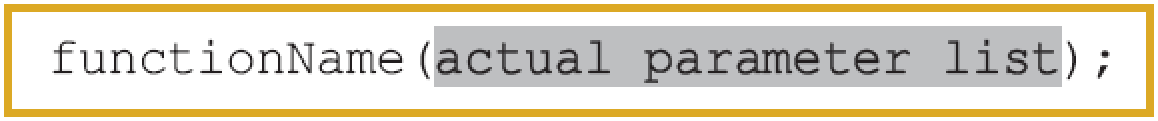
```
void functionName(formal parameter list)
{
      statements
}
```

- **Formal parameter list syntax:**



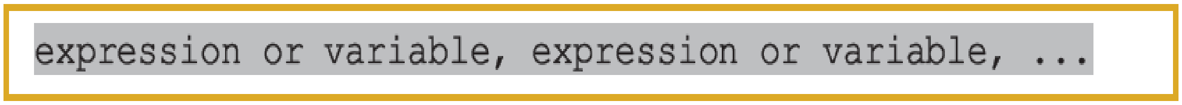
```
dataType& variable, dataType& variable, ...
```

- **Function call syntax:**

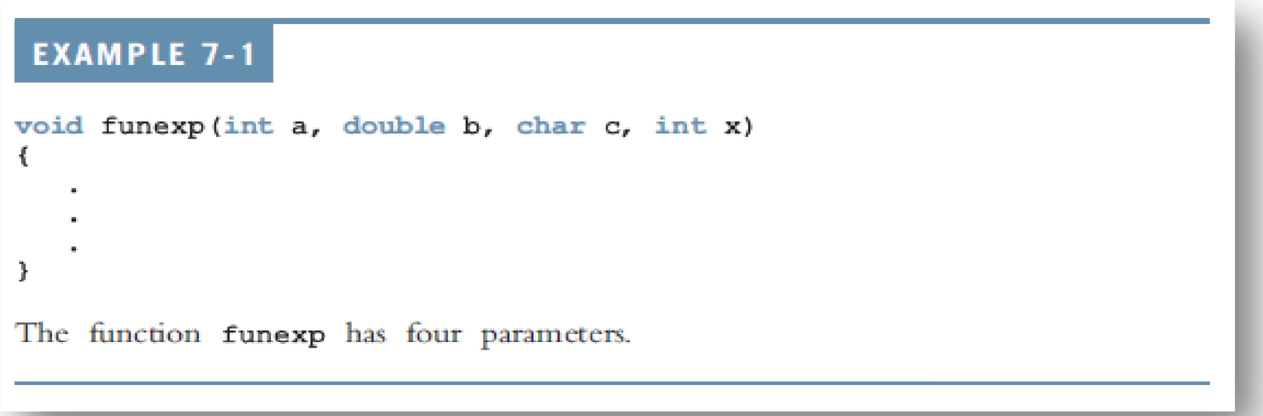


```
functionName(actual parameter list);
```

- **Actual parameter list syntax**



```
expression or variable, expression or variable, ...
```



EXAMPLE 7-1

```
void funexp(int a, double b, char c, int x)
{
   .
   .
   .
}
```

The function funexp has four parameters.

Menus Design using void functions (Programming Example 1)

```
#include <iostream>
using namespace std;
const double PI = 3.1419;
double rectangle(double l, double w);
double circle(double r);
double cylinder(double bR, double h);
void PrintMenu (); //function prototype
#include <iomanip>
int choice;
int main()
{
  double radius;
  double height;
  double length;
  double width;
   PrintMenu (); //function calling
    cout << fixed << showpoint << setprecision(2) << endl;
    while (choice != -1)
    {
       switch (choice)
        {
        case 1:
             cout << "Enter the length and the width "
                  << "of the rectangle: ";
            cin   >> length >> width;
            cout << endl;
            cout << "Area of Rectangle = "
                  << rectangle(length,width) << endl;
            break;
        case 2:
             cout << "Enter the radius of the circle: ";
            cin >> radius;
            cout << endl;
            cout << "Area = " << circle(radius)
                 << endl;
            break;
        case 3:
            cout << "Enter the radius of the base and the "
                 << "height of the cylinder: ";
            cin >> radius >> height;
            cout << endl;
            cout << "Volume = " << cylinder(radius, height)
                  << endl;
            break;
      default:
            cout << "Invalid choice!" << endl;
        }
       PrintMenu (); //function calling
    }
    return 0;
 }

                      // function definition
void PrintMenu ()
{
   cout << "This program can calculate the area of a rectangle, "
    << "the area of a circle, or volume of a cylinder." << endl;
    cout << "To run the program enter: " << endl;
    cout << "1: To find the area of rectangle." << endl;
    cout << "2: To find the area of a circle." << endl;
    cout << "3: To find the volume of a cylinder." << endl;
    cout << "-1: To terminate the program." << endl;
    cout << "Enter Your Choice\t";
    cin >> choice;
    cout << endl;
}
double rectangle(double l, double w)
{
    return l * w;
}
double circle(double r)
{
    return PI *r* r ;
}
double cylinder(double bR, double h)
{
    return PI * bR * bR * h;
}
```

Programming Example 2

```
//Program: Print a triangle of stars
#include <iostream>
using namespace std;
void printStars(int blanks, int starsInLine);
int main()
{
    int noOfLines;  //variable to store the number of lines
    int counter;    //for loop control variable
    int noOfBlanks; //variable to store the number of blanks
    cout << "Enter the number of star lines (1 to 20) "
         << "to be printed: ";                          //Line 1
    cin >> noOfLines;                                   //Line 2
    while (noOfLines < 0 || noOfLines > 20)             //Line 3
    {
        cout << "Number of star lines should be "
             << "between 1 and 20"<<endl;               //Line 4
        cout << "Enter the number of star lines "
             << "(1 to 20) to be printed: ";            //Line 5
        cin  >>  noOfLines;                             //Line 6
    }
    cout << endl << endl;                               //Line 7
    noOfBlanks = 30;                                    //Line 8
    for (counter = 1; counter <= noOfLines; counter++)  //Line 9
    {
        printStars(noOfBlanks, counter);                //Line 10
        noOfBlanks--;                                   //Line 11
    }
    return 0;                                           //Line 12
}
void printStars(int blanks, int starsInLine)
{
    int count;
    for (count = 1; count <= blanks; count++)           //Line 13
        cout << ' ';                                    //Line 14
    for (count = 1; count <= starsInLine; count++)      //Line 15
        cout << " *";                                   //Line 16
    cout << endl;
}
```