Lab8

Functions

Function

- A block of code which performs specific action.
- Runs only when called.
- Can be called many times(reusability of code).
- Also called method/procedure/sub-routine.
- General Syntax:

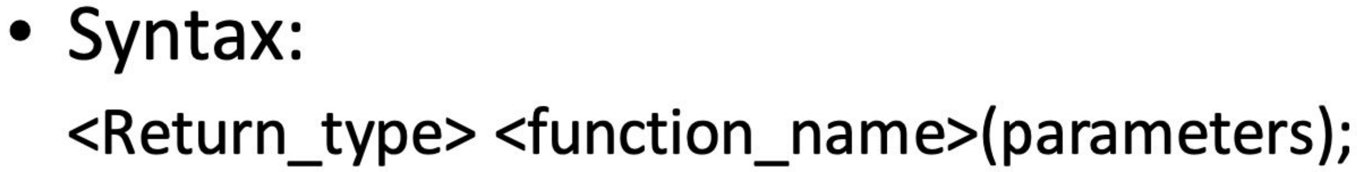
```
<Return type> function_name(parameters)
{
    //body of function
}
```

Function Definition

```
//template

<return type> <name> ( <parameters> )

{

   <function body>

   <return statement for return types other than void>

 }
```

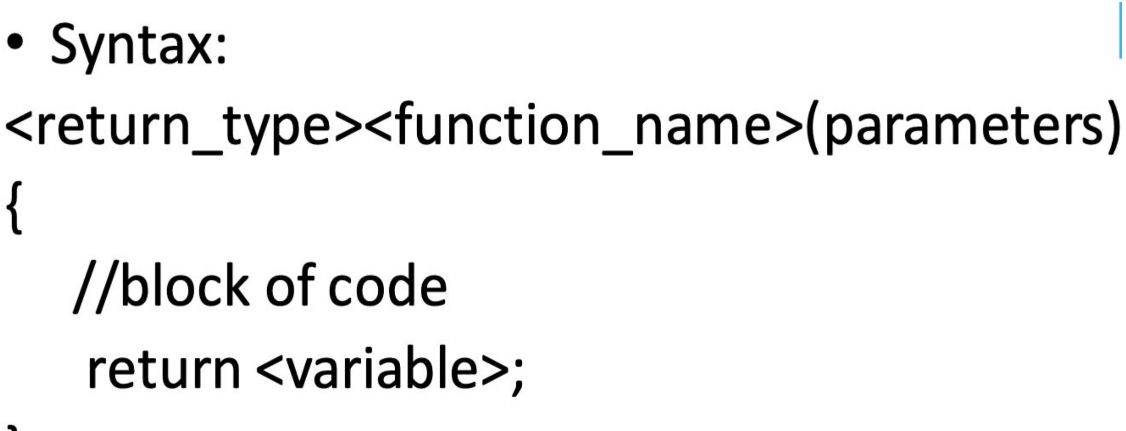
Function Definition and Declaration

A function declaration tells the compiler about a function's name, return type, and parameters. A function definition provides the actual body of the function. A declaration is also required if we’re defining the function after the main()



- Syntax:

```
<Return_type> <function_name>(parameters);
```



- Syntax:

```
<return_type><function_name>(parameters)
{
   //block of code
    return <variable>;
}
```

Return Type

Void functions: Does not return a function value

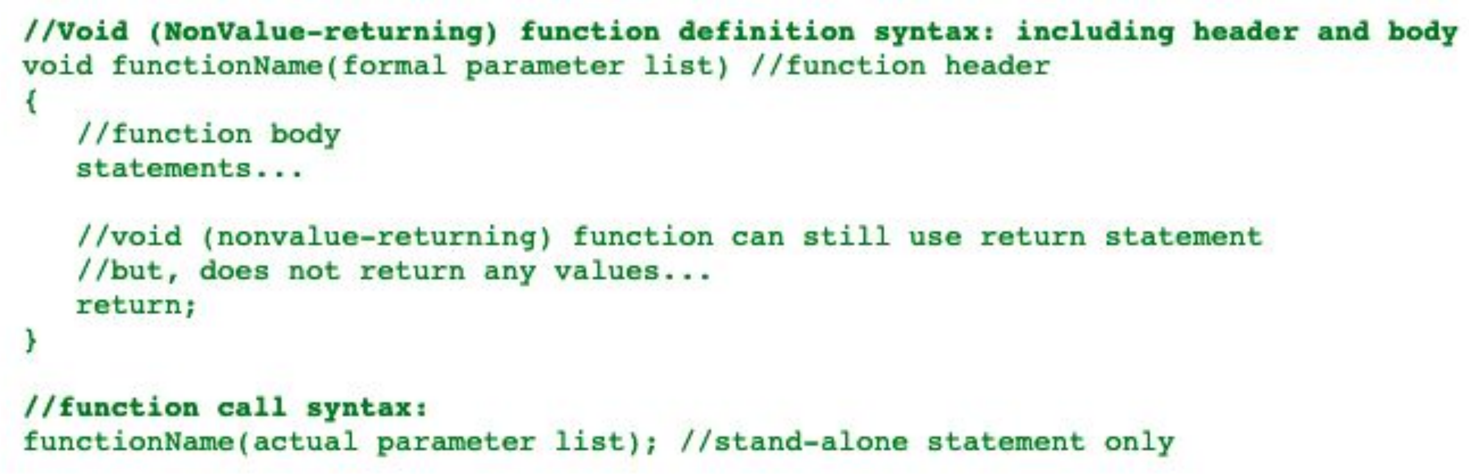
Non-void functions: Also known as value returning functions (returns value)

Similarities between void and non-void functions

- Both: require function definitions (i.e., headers and bodies)
- Both: definitions can be placed before or after function main()... though, it’s preferable to have it defined before main()
- Both: formal parameter list can be empty--though, parentheses still required
- Both: actual parameter list can use expression or variable, **but must match in "TON": type, order, number**

Differences

- Void function: does not have return type
- Uses keyword **void** in function header
- Call to void function is stand-alone statement



```
//Void (NonValue-returning) function definition syntax: including header and body
void functionName(formal parameter list) //function header
{
   //function body
   statements...

   //void (nonvalue-returning) function can still use return statement
   //but, does not return any values...
   return;
}

//function call syntax:
functionName(actual parameter list); //stand-alone statement only
```

Continued….

Conversely, value-returning function calls can be used in

- **output: e.g., cout statements**
- **Assignment**
- **arguments in other function calls**

Quick Exercise: Void or Non-void??

1. A function to find the cube of any int
2. A function that prints "Hello" 1000 times
3. A function that takes a string and a number and prints it took the screen that number of times
4. A function that calculates the nth fibonacci number
5. A function that takes a string called *phrase* and a file name as parameters. It writes that phrase to the file

Exercise

In this lab you'll make a single large program that uses several function definitions. Your main should have very little going on. In fact, I've provided your entire main below:

```
int main()
{
  run();
  return(0);
 }
```

Exercise

What your program will do is present the user with the following menu:

```
1) Count digits

2) Sum digits

3) Is Palindrome

4) Reverse

5) Exit

 Choice:
```

What is a Palindrome?

A palindrome is when a word, phrase or sequence reads in backwards the same way as in forward:

- MADAM
- 1221