

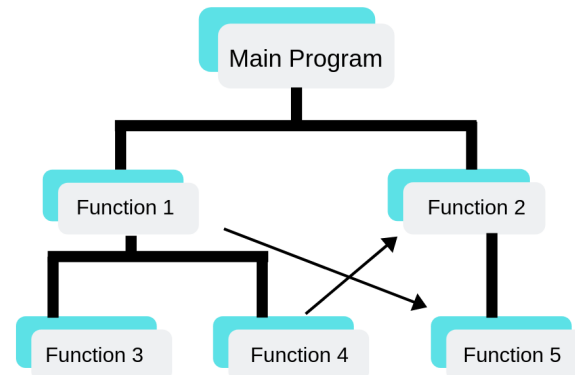


Fundamental Coding with C

Functions 1

Functions 1

- A function is a collection of statements that together performs a specific task.
- Some definition: A function is a named, independent section of code that performs a specific task.
- A complex problem is often easier to solve by dividing it into several smaller tasks each of which can be solved by itself.
- The functions optionally returns a value to the calling statement or/and receives values(s) from the calling statement.
- Dividing the problem into small tasks is called structured programming.





Functions 1

- Basically there are two categories of functions:
 - Predefined functions: available in standard libraries such as `stdio.h` or `string.h` in C.
 - User-defined functions: functions that programmers create for specific tasks within the program.

```
var scrollHeight =  
element.clientHeight + 0.02 * window.innerWidth  
window.scroll(0, scrollHeight);  
}
```



Functions 1

- Suppose you want to compute the square of 10 numbers. One programming solution is define 10 separate variables calculate the square of each and display it.
- What if the number of squares to be computed is increased to 30? It is not practically feasible to follow the traditional method. This is where the role of functions comes into play.
- The above-mentioned problem can easily be resolved by defining a function to compute the square of a number taking one parameter into consideration, that is, the number.



Functions 1

- **Function Declaration in C**

```
return_type function_name (parameter list)
{
    function_statement(s)
}
```

- **return_type** – A function may return a value. The return_type is the data type of the value the function returns. Some functions perform the desired operations without returning a value. In this case, the return_type is the keyword void.
- **function_name** – This is the actual name of the function. The function name and the parameter list together constitute the function signature.



Functions 1

- **Function Declaration in C**

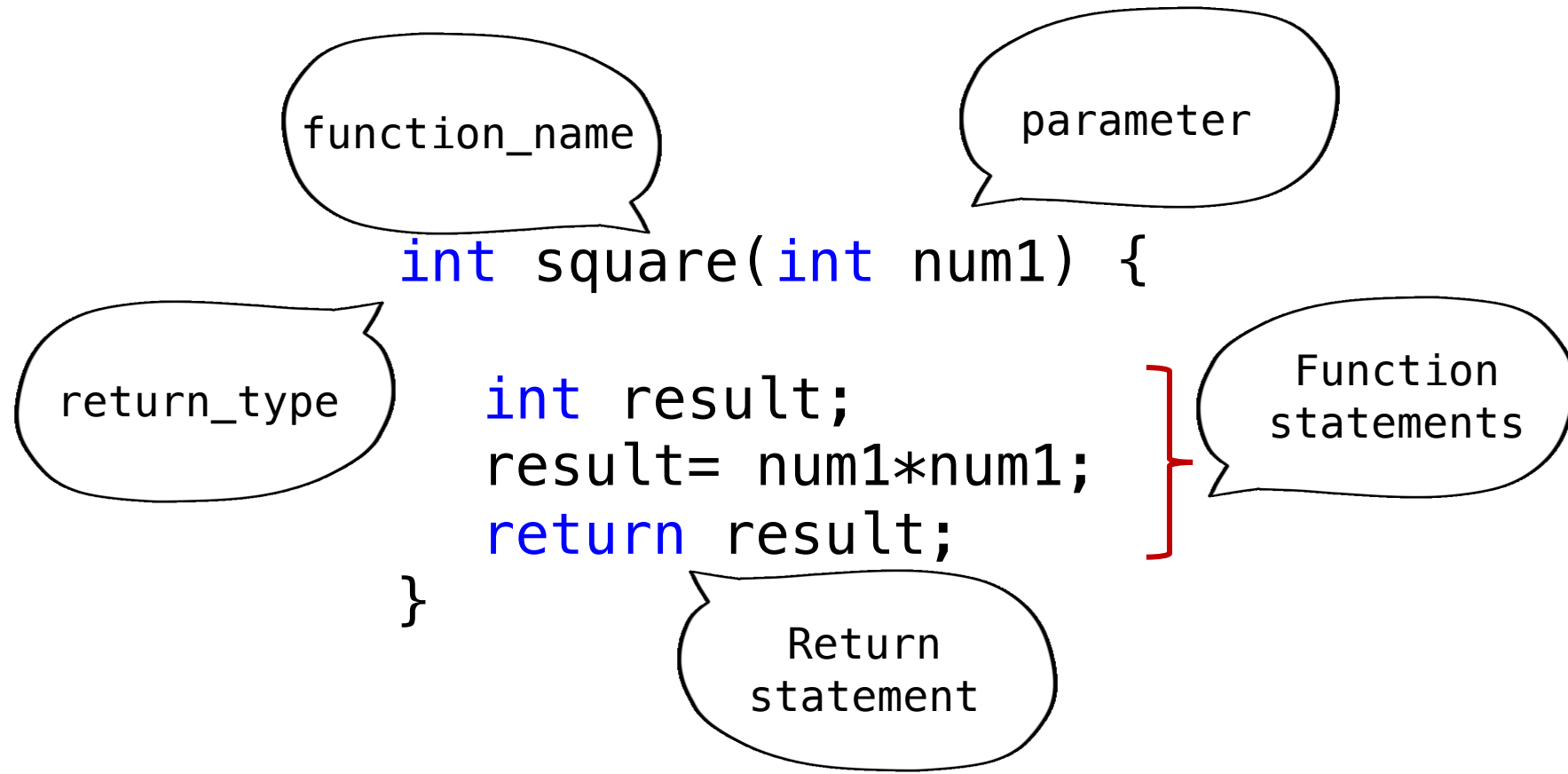
```
return_type function_name (parameter list)
{
    function_statement(s)
}
```

- **parameter list** – The parameter list refers to the type, order, and number of the parameters of a function. Parameters are optional; that is, a function may contain no parameters.
- **function_statement(s)** – The function statement(s) contains a collection of statements that define what the function does.



Functions 1

- Function declaration example in C





Functions 1

- Function use example in C

```
int number=12;  
printf("Scuare of %d is %d \n",number, square(number))
```

Function call



Fundamental
Coding with C

It's time to try

<https://repl.it/languages/c>





- **Function example in C**

```
#include <stdio.h>

int square(int num1) {

    int result;
    result= num1*num1;
    return result;

}

int main(void) {
int number=12;
printf("Square of %d is %d \n",number,
square(number));
return 0;
}
```