Tassadaq Hussain

Structure

A data structure is a group of data elements grouped together under one name.
These data elements, known as members, can have different types and different lengths.

```
struct name{
Data types
} variable_struct;
```

Functions

Functions

Functions

Functions allow to structure programs in segments of code to perform individual tasks.

return_type function_name (parameter1, parameter2, ...)
{ statements }

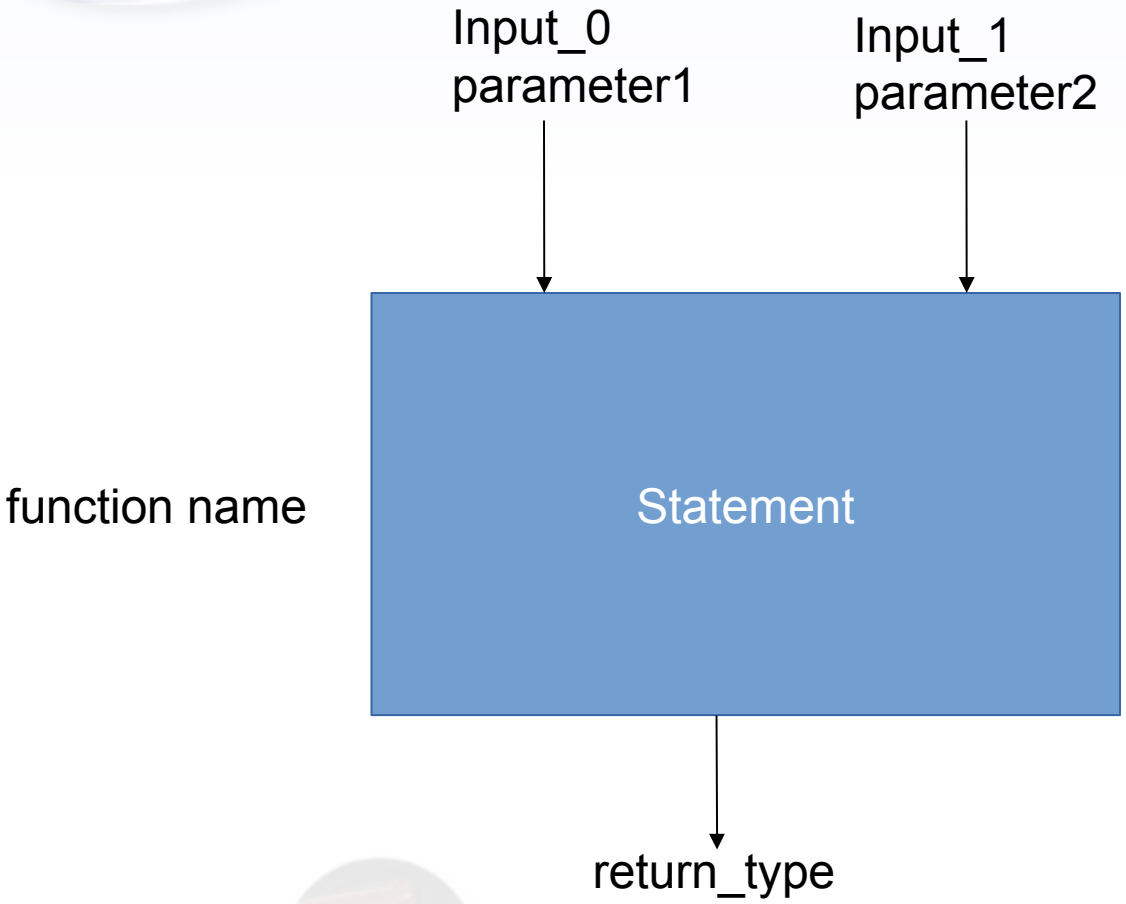
Return_type
The return type of the value returned by the function.

Function_name
is the identifier by which the function can be called.

Parameters (as many as needed):
Each parameter consists of a type followed by an identifier, with each parameter being separated from the next by a comma. Each parameter looks very much like a regular variable declaration (for example: int x), and in fact acts within the function as a regular variable which is local to the function. The purpose of parameters is to allow passing arguments to the function from the location where it is called from.

Statements
is the function's body. It is a block of statements surrounded by braces { } that specify what the function actually does.

return_type function_name (parameter1, parameter2, ...)
{ statements }



Calling Function

```
void main()
{
int x;
x=function_name(input_0, input_1);
}
```

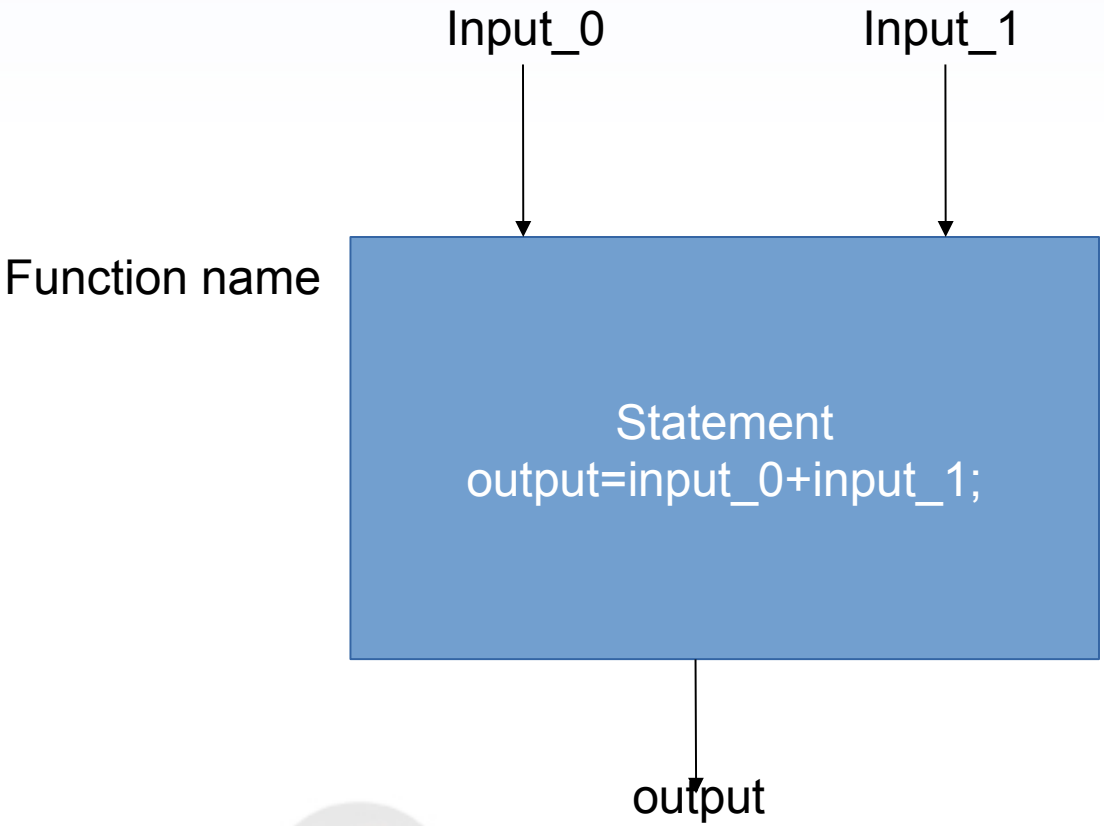
A Simple Function with no output

```
void print_names(string name)
{
cout<<name<<endl;
}
void main()
{
print_name(“Iftikhar”);
}
```

Simple Function Program with Return

```
int addition (int a, int b)
{
  int r;
  r=a+b;
  return r;
}
```

```
int main ()
{
  int z;
  z = addition (5,3);
  cout << "The result is " << z;
}
```


Input_0
Input_1
Function name
Statement
output=input_0+input_1;
output