Functions — Review.

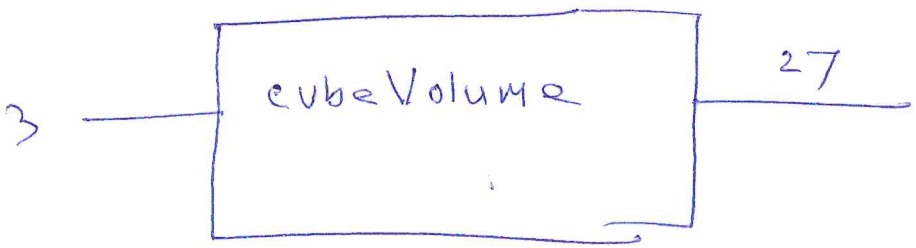
A function is a sequence of instructions with a name. For example, pow function contain instructions to compute power of x^y.

```
int main ()
{
    double result = pow(2,3);
}
```

function

A program to calculate the volume a cube using user defined function.

- Suppose you want to create a function named cubeVolume, which will calculate the volume of a cube.
- This function takes sidelength as an argument and returns volume of the cube.



Syntax for a function that returns a value

```
returnType functionName (parameterType parameterName...);
int main()
{
    functionName (parameterValue...) ;
}
returnType functionName (parameterType parameterName...)
{
  statement body
}
```

Function Call

Function Definition

```
double cubeVolume (double sidelength);
int main()
{
  cubeVolume (2);
}
double cubeVolume (double sidelength)
{
  double volume = sidelength * sidelength * sidelength ;
  return volume;
}
```

Function declaration

Function call

Function definition

Function Declaration: A function declaration tells you all you need to know to write a call to the function. Function declarations are placed before the main part of your program.

```
returnType functionName (parameterType parameterName...);
```

Function call: Function call is a way of calling a function. When a function is called, its argument (actual parameters) are copied to the formal parameters.

```
functionName (parameterValue...) ;
```

Function Definition: A function definition describes how the function computes the value it returns. A function consists of a function header followed by a function body.

```
returnType functionName (parameterType parameterName...)
{
  statement body
}
```

Using programmer defined function, write a program to determine the volume of a cube. The equation for the volume is:

V=sidelength*sidelength*sidelength

Volume

8 .

```
#include <iostream>

using namespace std;
double cubeVolume (double sidelength);

int main()
{
    double volume = cubeVolume (2);
    cout<<"The volume of the cube is: "<<volume;
}

    double cubeVolume (double sidelength)
{
    double volume = sidelength * sidelength * sidelength ;
    return volume;
}
```

Function declaration
~~Definition~~

Function call

Function Definition

OUTPUT

```
The volume of the cube is: 8
```

Practice Problem 1: What is the output of following program (when embedded in a complete program?)

```
double mystery (double x, double y);
int main()
{
   double a = 5;
   double b = 7;
   cout<<mystery(a,b);
}

double mystery (double x, double y)
{
   double z = x + y;
   z= z/2.0;
   return z;
}
```

5 + 7 = 12

$z = \frac{12}{2.0} = 6.0$

6.0

Output .

6.0

Practice Problem 2: What is the output of following program (when embedded in a complete program?)

```
int mystery (int x)
int main()
{
   int a = 4;
   cout<< (mystery(a+1));
}

int mystery (int x)
{
   int y = x * x;
   return y;
}
```

5

y = 5 * 5 = 25

OUTPUT

25.

Boolean Type Return Value

What is the output of following program (when embedded in a complete program?)

```
bool mystery (int n);
int main()
{
    int n =6;
    bool var=mystery (n);
    if (var)
    {
        cout<<"Number is even";
    }
}

bool mystery (int n)
{
    if (n%2==0)
    {
        return true;
    }
    else
    {
        return false;
    }
}
```

var
true.

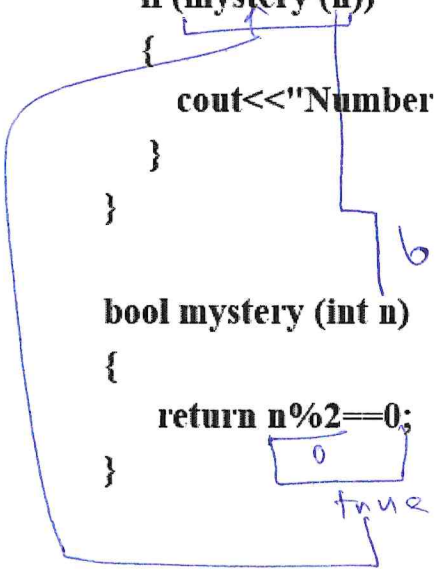
OUTPUT.

Number is even.

What is the output of following program (when embedded in a complete program?)

```
bool mystery (int n);
int main()
{
   int n =6;
   if (mystery (n))
   {
      cout<<"Number is even";
   }
}

bool mystery (int n)
{
    return n%2==0;
}
```



OUTPUT

Numbe is even .

Write a C++ program to add two given integer numbers.

```
#include <iostream>
using namespace std;

int main() // main code
{
   int a=5, b=4, c;
   c = a+b;
   cout<<"Result:"<<c;
   return 0;
}
```

OUTPUT

Result : 9 .

Write a C++ program to add two given integer numbers using user defined function.

```
#include <iostream>
using namespace std;

int add (int x, int y); // function declaration
int main() // main code
{
   int a=5, b=4, c;
   c = add(a, b); // function call
   cout<<"Result:"<<c;
   return 0;
}

int add (int x, int y) // function definition
{
   return (x + y);
}
```

c = 9.

OUTPUT

Result : 9

5 4

5+4

ARMv7

```
100   MOV   R1, #5        // a=5
104   MOV   R2, #4        // b=4
108   BL    addition
112   B     exit

addition:
      ADD   R9, R1, R2    // c=a+b=9.
      BX    LR
108
Exit :
```