Computer Programming (a) - E1123

(Fall 2021-2022)

Lecture 6

Functions

INSTRUCTOR

DR / AYMAN SOLIMAN

➢ Contents

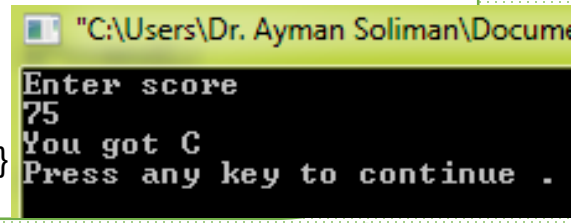
1) Examples
2) Introduction
3) Predefined Functions
4) user-defined Functions
5) Function Call

➢ Switch Case Example

Write a C++ program to enter the score degree in programming language, then the output will be as the following table:

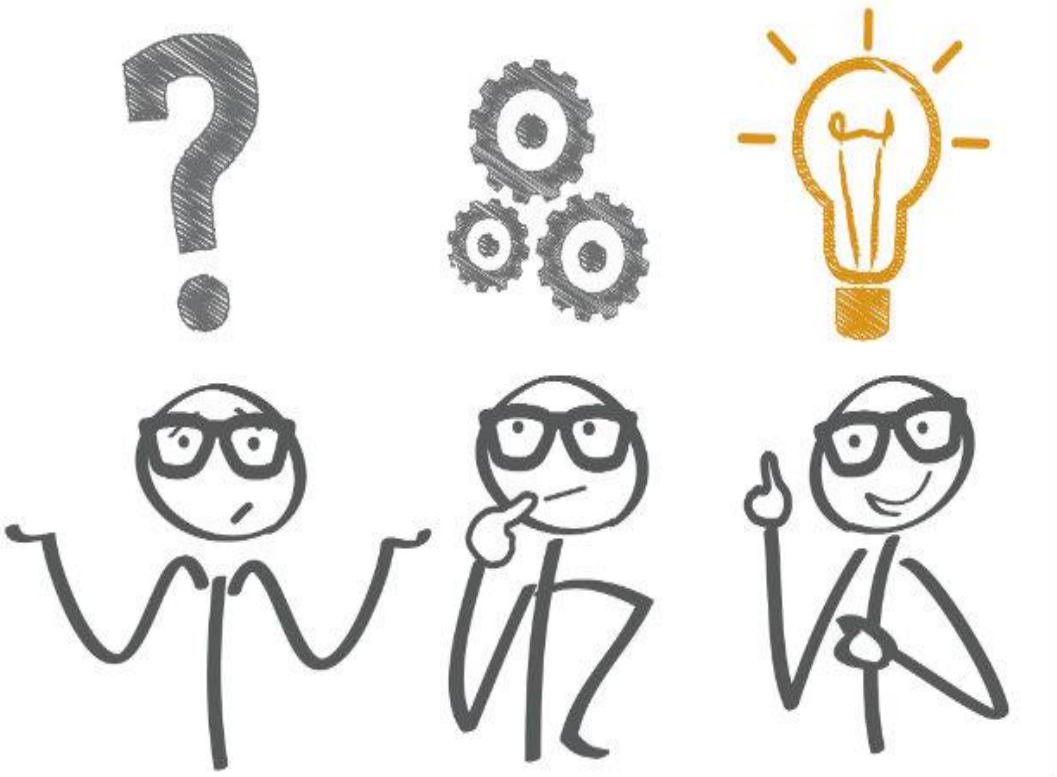
Score	Output
100	Perfect
90 - 99	A
80 – 89	B
70 – 79	C
60 – 69	D
50 – 59	E
0 – 49	F

```
int score;
cout<<"Enter score"<<endl;
cin>>score;

switch(score){
 case 100:
    cout<<"Your score is Perfect"<<endl;
 break;
 case 90 ... 99:
    cout<<"You got A"<<endl;
 break;
 case 80 ... 89:
    cout<<"You got B"<<endl;
    break;
 case 70 ... 79:
    cout<<"You got C"<<endl;
    break;
 case 60 ... 69:
    cout<<"You got D"<<endl;
    break;
 case 50 ... 59:
    cout<<"You got E"<<endl;
    break;
 case 0 ... 49:
    cout<<"You got F"<<endl;}
```



➢ Example 2

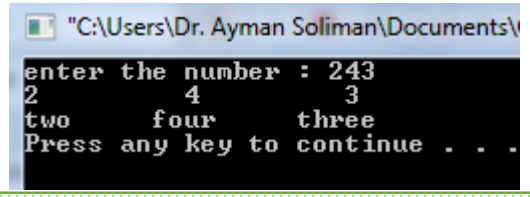
Write a C ++ program to enter a three numbers and print each number individually and in words using switch case method.



```
enter the number : 537
5     3     7
five three seven
Press any key to continue . . .
```

➢ Example 2 - solution

```
#include <iostream>
#include <string>
using namespace std;
int main()
{
        int x,a1,a2,b1,c1;
        string a11,b11,c11;
        cout<<"enter the number : ";
        cin>>x;
        a1=x/100;
        a2=x%100;
        b1=a2/10;
        c1=a2%10;
        cout<<a1<<"    "<<b1<<"    "<<c1<<endl;
```




```
switch (a1)
{case 0:
 a11="zero";
 break;
 case 1:
 a11="one";
 break;
 case 2:
 a11="two";
 break;
 case 3:
 a11="three";
 break;
 case 4:
 a11="four";
 break;
 case 5:
 a11="five";
 break;
 case 6:
 a11="six";
 break;
 case 7:
 a11="seven";
 break;
 case 8:
 a11="eight";
 break;
 case 9:
 a11="nine";
 break;        }
```

```
switch (b1)
 {case 0:
 b11="zero";
 break;
 case 1:
 b11="one";
 break;
 case 2:
 b11="two";
 break;
 case 3:
 b11="three";
 break;
 case 4:
 b11="four";
 break;
 case 5:
 b11="five";
 break;
 case 6:
 b11="six";
 break;
 case 7:
 b11="seven";
 break;
 case 8:
 b11="eight";
 break;
 case 9:
 b11="nine";
 break;        }
```

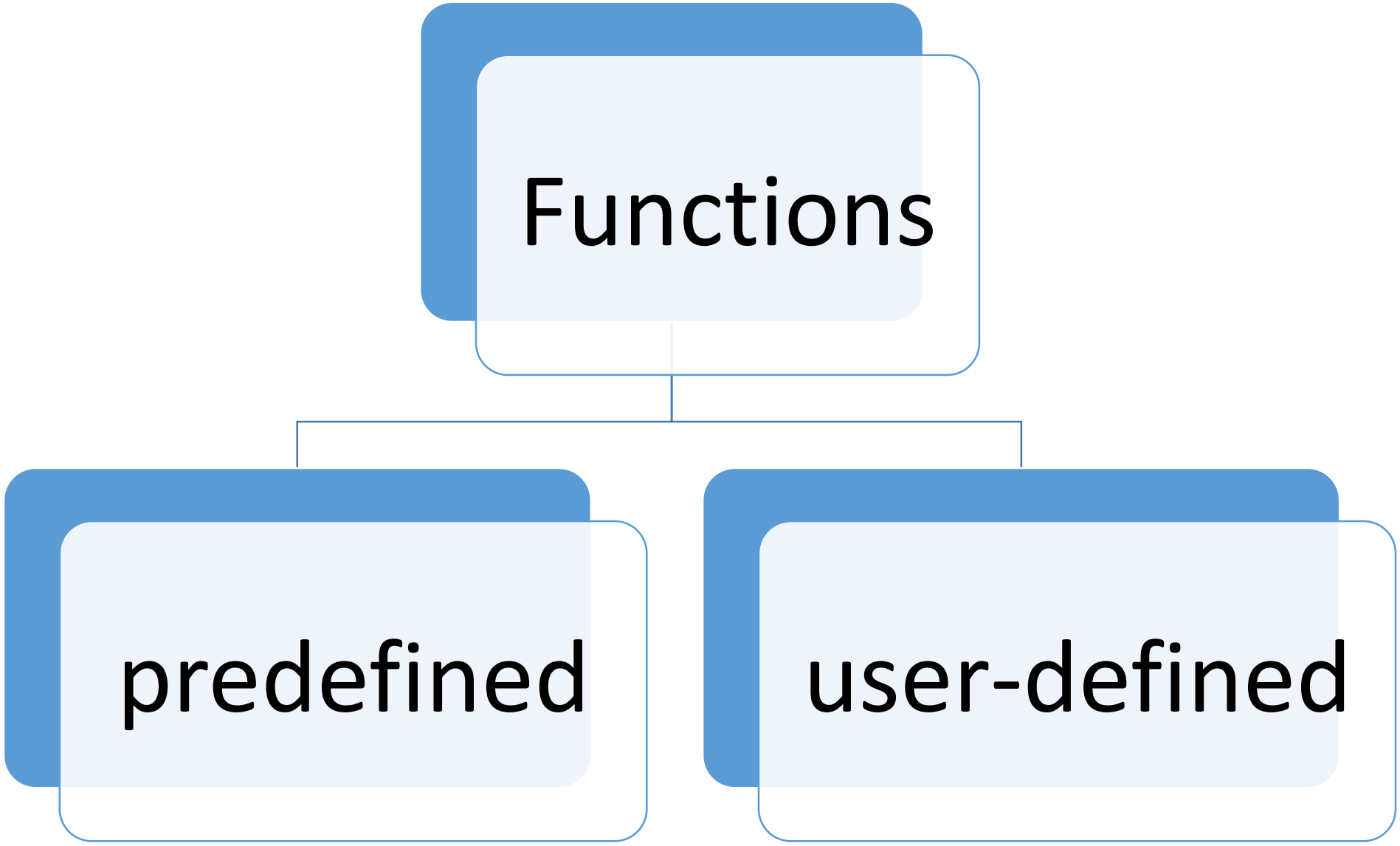
```
switch (c1)
 {case 0:
  c11="zero";
  break;
  case 1:
  c11="one";
  break;
  case 2:
  c11="two";
  break;
  case 3:
  c11="three";
  break;
  case 4:
  c11="four";
  break;
  case 5:
  c11="five";
  break;
  case 6:
  c11="six";
  break;
  case 7:
  c11="seven";
  break;
  case 8:
  c11="eight";
  break;
  case 9:
  c11="nine";
  break;        }
```

```
cout<<a11<<"    "<<b11<<"    "<<c11<<endl;
return 0;
}
```

➢Introduction

In this Lecture, you will:

- Learn about standard (predefined) functions and discover how to use them in a program
- Learn about user-defined functions
- Examine value-returning functions, including actual and formal parameters
- Explore how to construct and use a value-returning, user-defined function in a program


Functions
predefined
user-defined

➢Introduction

- In algebra, a function is defined as a rule or correspondence between values, called the function's arguments, and the unique value of the function associated with the arguments

  - If f(x) = 2x + 5, then f(1) = 7, f(2) = 9, and f(3) = 11

    - 1, 2, and 3 are arguments

    - 7, 9, and 11 are the corresponding values

➢Predefined Functions

Some of the predefined mathematical functions are:

`pow(x,y)` calculates x^y

`pow(2,3) = 8.0`

Returns a value of type `double`

`x` and `y` are the parameters (or arguments)

The function has two parameters

`sqrt(x)` calculates the nonnegative square root of `x`, for `x >= 0.0`

`sqrt(2.25)` is `1.5`

Type `double`

Function	Description	Argument type	Return type	Header file
sqrt	Square root	double	double	cmath
pow	Powers	double	double	cmath
abs	Absolute value for int	int	int	cstdlib
exit	End program	int	void	cstdlib
assert	Abort program with message	boolean	void	assert

➢Predefined Functions

		Built-in Functions		
abs()	dict()	help()	min()	setattr()
all()	dir()	hex()	next()	slice()
any()	divmod()	id()	object()	sorted()
ascii()	enumerate()	input()	oct()	staticmethod()
bin()	eval()	int()	open()	str()
bool()	exec()	isinstance()	ord()	sum()
bytearray()	filter()	issubclass()	pow()	super()
bytes()	float()	iter()	print()	tuple()
callable()	format()	len()	property()	type()
chr()	frozenset()	list()	range()	vars()
classmethod()	getattr()	locals()	repr()	zip()
compile()	globals()	map()	reversed()	__import__()
complex()	hasattr()	max()	round()	
delattr()	hash()	memoryview()	set()	

➢Predefined Functions

```
#include <iostream>
#include <cmath> // for sqrt and pow
using namespace std;

int main()
{
    double number, squareRoot;
    cout << "Enter a number: ";
    cin >> number;
    // sqrt() is a library function to calculate square root
    squareRoot = sqrt(number);
    double power=pow(number,3);
    cout << "Square root of " << number << " = " << squareRoot;
    cout << endl;
    cout <<  number << " ^ 3 = " << power;
    return 0;
}
```

```
Enter a number: 2.5
Square root of 2.5 = 1.58114
2.5 ^ 3 = 15.625
```

➢Predefined Functions

```
#include <iostream>
#include <cmath>
#include <cctype>
#include <cstdlib>

using namespace std;

int main()
{
    int    x;
    double u, v;


    cout << "Line 1: Uppercase a is "
         << static_cast<char>(toupper('a'))
         << endl;                                          //Line 1

    u = 4.2;                                               //Line 2
    v = 3.0;                                               //Line 3
    cout << "Line 4: " << u << " to the power of "
         << v << " = " << pow(u, v) << endl;               //Line 4

    cout << "Line 5: 5.0 to the power of 4 = "
         << pow(5.0, 4) << endl;                           //Line 5

    u = u + pow(3.0, 3);                                   //Line 6
    cout << "Line 7: u = " << u << endl;                   //Line 7

    x = -15;                                               //Line 8
    cout << "Line 9: Absolute value of " << x
         << " = " << abs(x) << endl;                       //Line 9

    return 0;
}
```

```
Line 1: Uppercase a is A
Line 4: 4.2 to the power of 3 = 74.088
Line 5: 5.0 to the power of 4 = 625
Line 7: u = 31.2
Line 9: Absolute value of -15 = 15
```

➢user-defined Functions

Syntax:

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

Function type is also called the data type or return type

➢Function Call

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

➢Return Statement

Once a value-returning function computes the value, the function returns this value via the return statement

- It passes this value outside the function via the return statement
- The return statement has the following syntax:

  - ➢ In C++, return is a reserved word
  - ➢ When a return statement executes, function immediately terminates
  - ➢ Control goes back to the caller
  - ➢ When a return statement executes in the function main, the program terminates

➢user-defined Functions

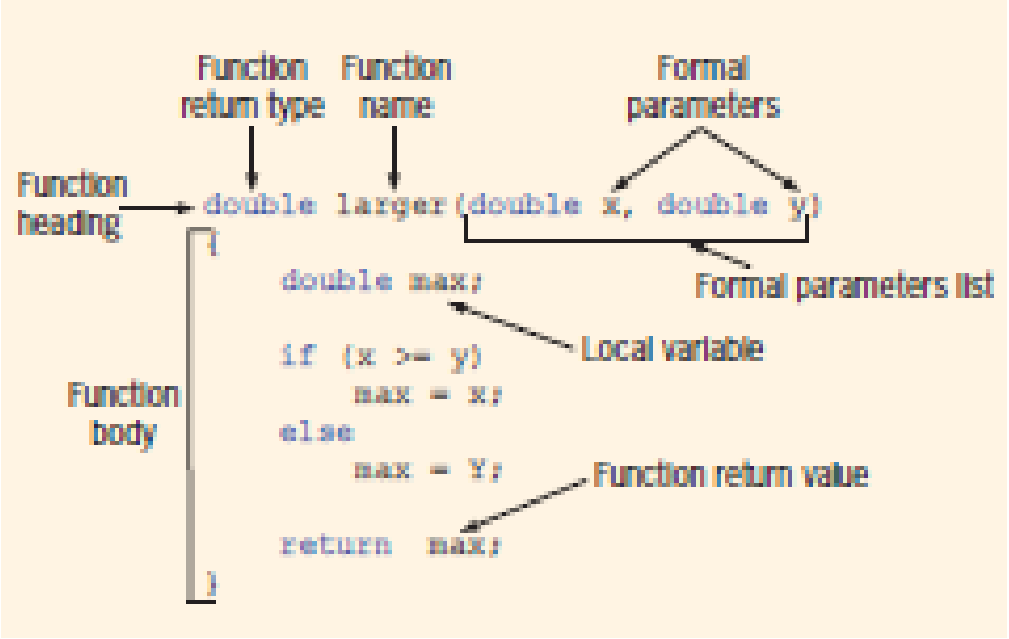
```
double larger ( double x, double y)
{
        double max;

        if ( x >= y )
            max = x;
        else
            max = y;

        return max;
}
```

You can also write this function as follows:

```
double larger ( double x, double y)
{
        if ( x >= y )
            return x;
        else
            return y;
}
```

You can also write this function as follows:

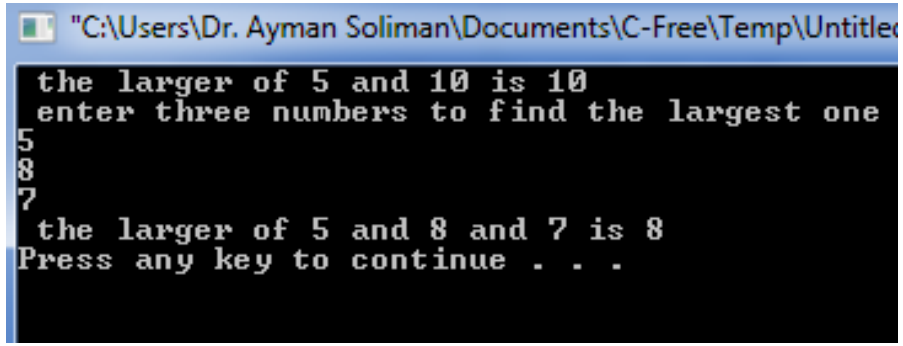
```
double larger ( double x, double y)
{
        if ( x >= y )
            return x;
        return y;
}
```



➢user-defined Functions

```
// program: find largest of two or three numbers

# include <iostream.h>
double larger ( double x, double y);
double comparethree ( double x, double y, double z);
int x,y,z;
int main()
{
     cout<< " the larger of 5 and 10 is "<< larger (5,10) << endl;

     cout<< " enter three numbers to find the largest one\n";
     cin>> x >> y>> z;

     cout<< " the larger of "<< x<<" and "<<y<<" and "<< z<<" is “
          << comparethree(x,y,z) << endl;
          return 0;
}
```

```
double larger ( double x, double y)
{
          if ( x >= y )
               return x;
          else
               return y;
}

double comparethree ( double x, double y, double z)
{
          return larger (x, larger(y,z));
}
```



➢user-defined Functions

```
# include <iostream.h>
// function definition
Void welcome()
{
        string name;
        cout <<“ enter your first name: “;
        cin >> name;
        cout << “Hey “<< name << “ ! ”;
}

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

```
enter your first name: Ayman

Hey Ayman!
```

➢user-defined Functions

The **return type** is the type declared before the function name. Note that the return type does not indicate what specific value will be returned. It only indicates what type of value will be returned.

```
# include <iostream.h>
// function definition
double return_value()
{
        return 3.2;
}

int main()
{
         cout<< return_value() <<endl;
        return 0;
}
```

```
3.2
```

```
# include <iostream.h>
// function definition
int return_value()
{
        return 3.2;
}

int main()
{
         cout<< return_value() <<endl;
        return 0;
}
```

```
3
```

➢user-defined Functions

```
# include <iostream.h>
// function definition
char return_value()
{
        return ‘C’;
}

int main()
{
         cout<< return_value() <<endl;
        return 0;
}
```

C

```
# include <iostream.h>
// function definition
int return_value()
{
        return ‘C’;
}

int main()
{
         cout<< return_value() <<endl;
        return 0;
}
```

67

Thank you