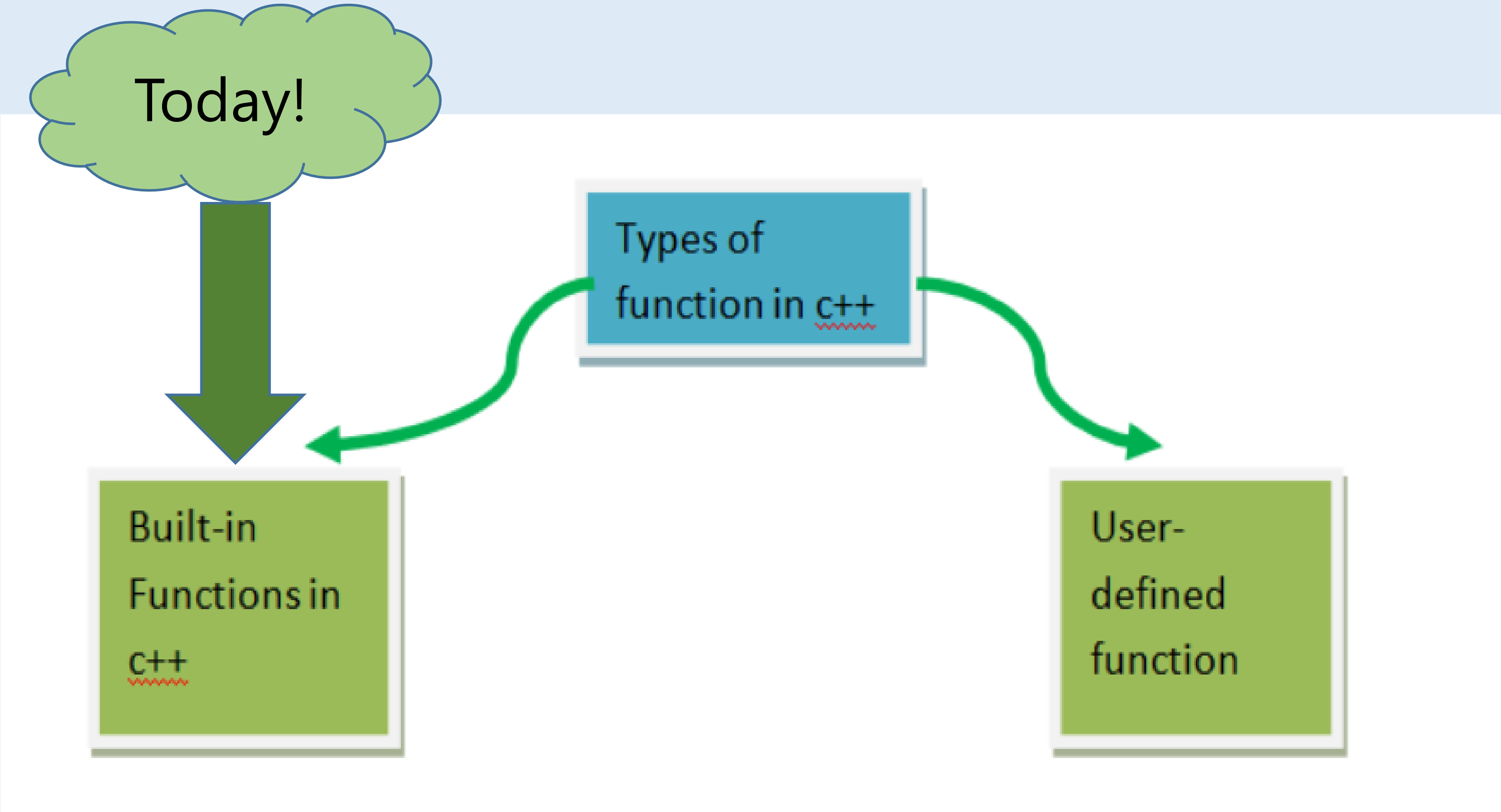
Lab #6

Functions (1)

Structured Programming 2017/2018

Today’s Lab

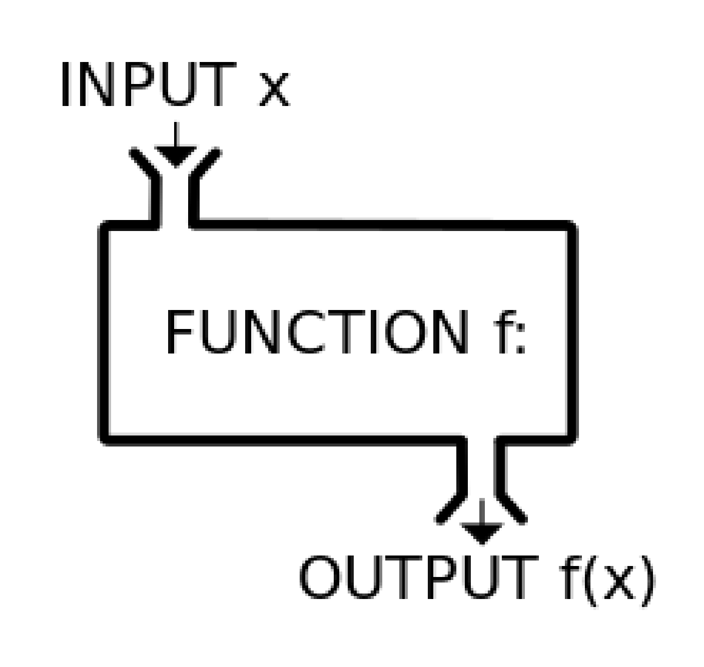

Today!
Types of function in c++
Built-in Functions in c++
User-defined function

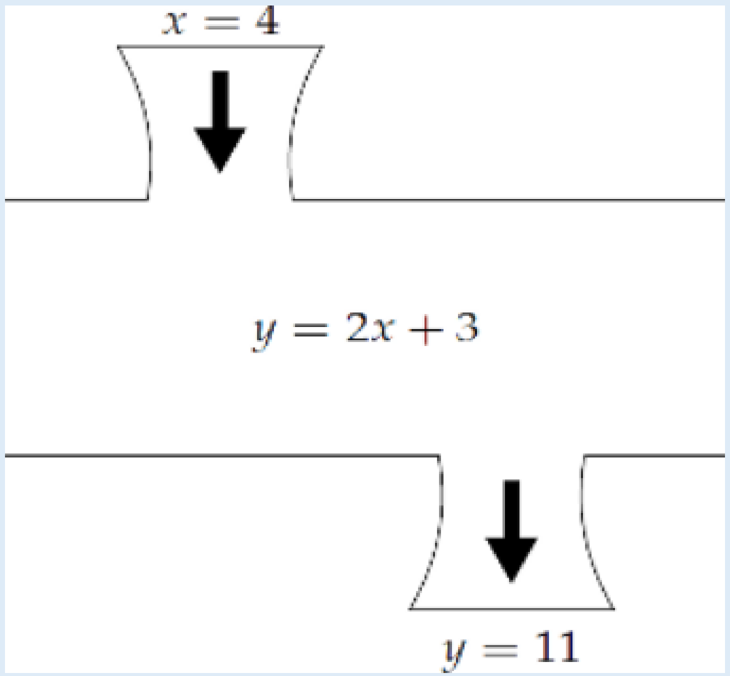
What is a function

- A large program is divided into many subprograms.
- The subprogram is called as a function.
- A function is a group of statements that together perform a **ONE SINGLE** task.
- Any program at least has one function which is the main() function.

What is a function

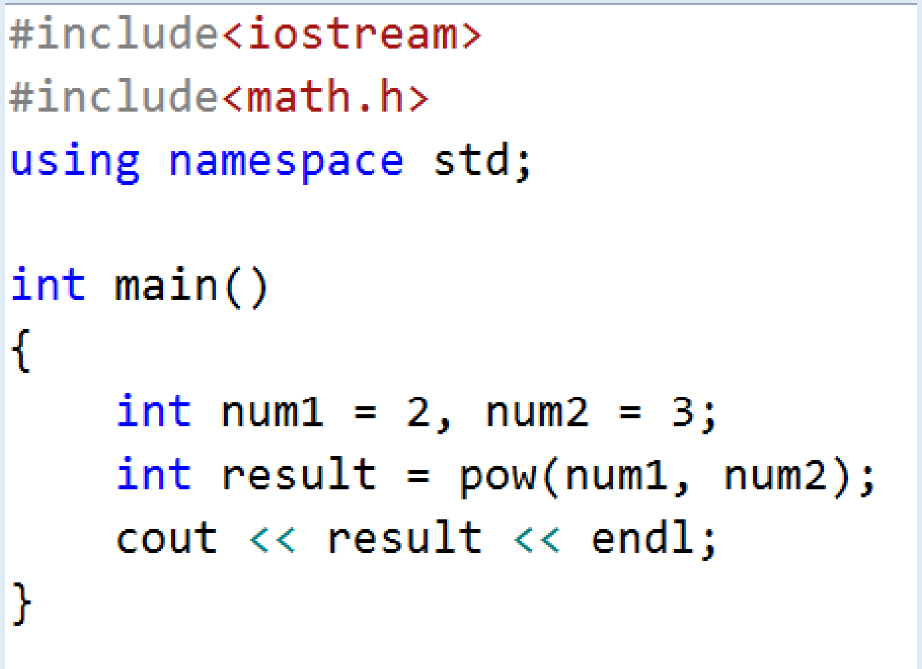
- A function takes input (parameters) and performs a single task (function body) and returns an output (return value).





Built-In (Predefined) functions

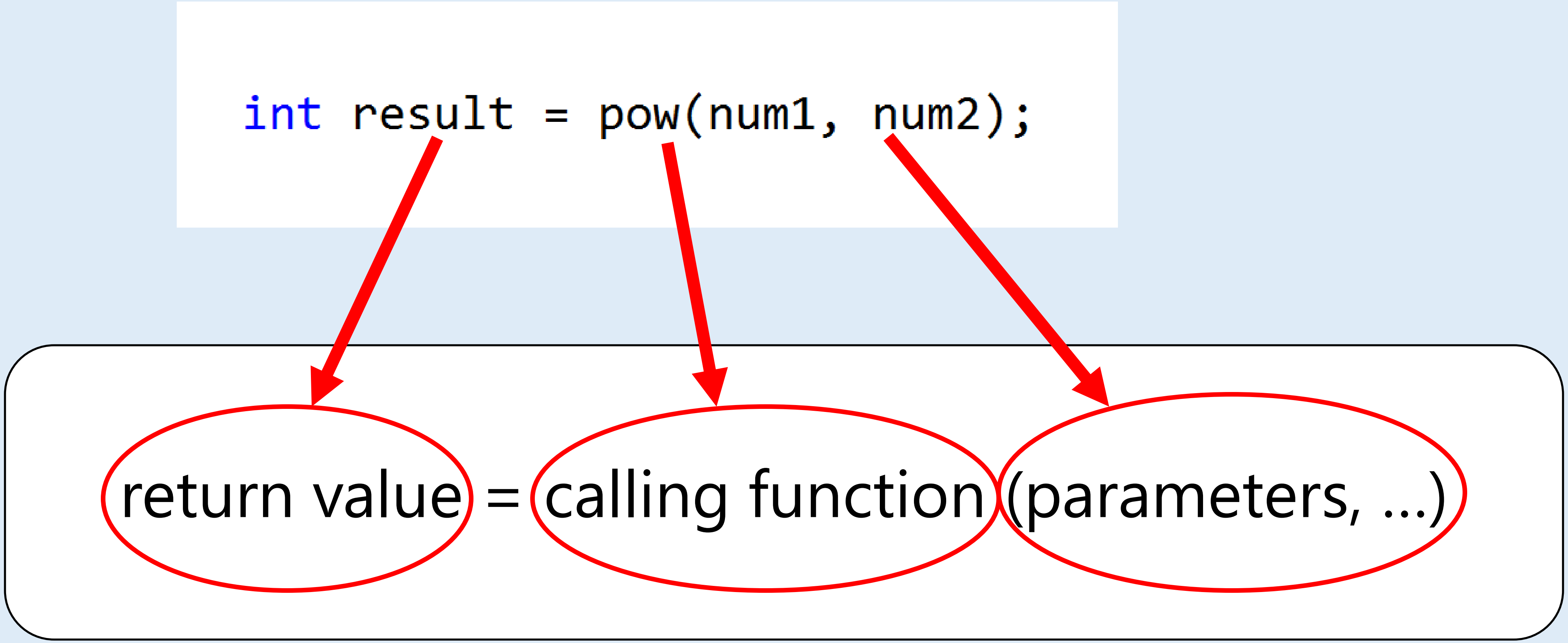
- These are built-in functions (already defined by c++ language) to handle tasks such as mathematical computations, I/O processing, string handling.
- These functions are defined in the header file. When you include the header file, these functions are available for use.
- Ex:



```cpp
#include<iostream>
#include<math.h>
using namespace std;

int main()
{
    int num1 = 2, num2 = 3;
    int result = pow(num1, num2);
    cout << result << endl;
}
```

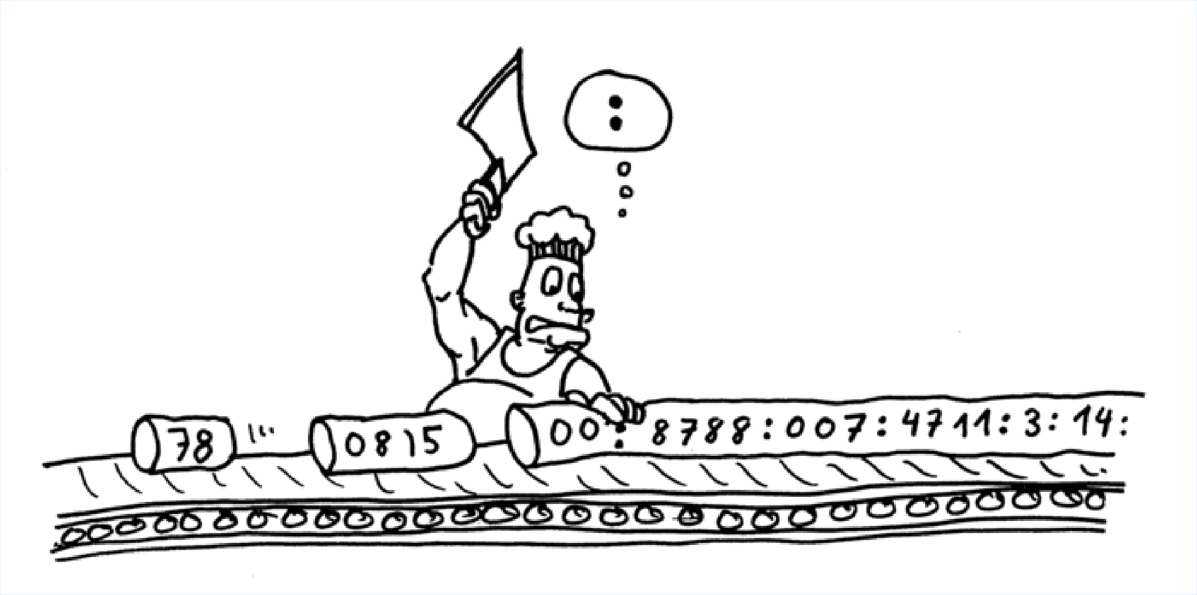
Built-In (Predefined) functions



```
int result = pow(num1, num2);
```

return value = calling function (parameters, ...)

Exercises!



Swap Function Example

- Write a program that reads the data of two students into two separate objects then swap their values using built-in function. Each student should have Id and Name properties

Sample of Execution:

```
Enter First Id: 1
Enter First Name: Ahmed
Enter Second Id: 2
Enter Second Name: Mona

First Student Data: 2 – Mona
Second Student Data: 1 - Ahmed
```

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

struct Student {
int Id;
char Name[10];
};

int main()
{

Student std1, std2;
cout << "Enter First Id : ";
cin >> std1.Id;
cout << "Enter First Name : ";
cin >> std1.Name;
cout << "Enter Second Id : ";
cin >> std2.Id;
cout << "Enter Second Name : ";
cin >> std2.Name;

swap(std1, std2);

cout << "First Student Data : " << std1.Id << " - " << std1.Name <<
endl;
cout << "Second Student Data : " << std2.Id << " - " << std2.Name
<< endl;

system("Pause");
return 0;
}
```

Sort Function Example

- Write a program that reads an array of 5 elements from the user and sorts them in order using built-in functions.

Sample of Execution:

```
Enter Elements: 1 7 6 8 0

Sorted Elements: 0 1 6 7 8
```

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

int main()
{
const int size = 5;
int arr[size];

cout << "Enter Elements: ";
for (int i = 0; i < size; i++)
cin >> arr[i];

sort(arr, arr + size);

cout << "Sorted Elements: ";
for (int i = 0; i < size; i++)
cout << arr[i] << "  ";

cout<<endl;
return 0;
}
```

Ready… Steady… Code!

Math Example

- For a quadratic equation ax2+bx+c = 0 where a, b and c are coefficients.
- Write a program that reads from the user a, b and c and evaluates the value of x by the following equation:

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

- Sample of Execution:

```
Enter A: 1
Enter B: -4
Enter C: 3

Output: x = 3, x = 1
```

```
#include<iostream>
#include<math.h>
using namespace std;

int main()
{
float a, b, c;

cout << "Enter a: ";
cin >> a;
cout << "Enter b: ";
cin >> b;
cout << "Enter c: ";
cin >> c;
float bSquare = pow(b, 2);
float sqrtVal = sqrt(bSquare - (4 * a*c));

float x1 = (-b + sqrtVal) / (2 * a);
float x2 = (-b - sqrtVal) / (2 * a);

cout << "x1 = " << x1 << " ,x2 = " << x2 << endl;

return 0;
}
```

Password Validation Example

- Write a program that validates a password entered by the user without displaying it. The password should contain at least one of these special characters ($, % , _ , #, @) and at least one number.

- Hints: - library: conio.h, built-in function _getch() → returns entered char

  - use two boolean variables isNum, isSpecial

  - use The ASCII code of numbers 0-9 is 48-57

<u>Sample of Execution:</u>

```
Enter Password: abc
Output: Not valid

Enter Password: abc_1
Output: Valid
```

```
#include<iostream>
#include<conio.h>
using namespace std;

int main()
{
char ch, password[10];
bool isSpecial = false, isNum = false;

cout << "Enter Password: ";
ch = _getch();
```

```cpp
while (ch != '\r')
{
if (ch == '$' || ch == '%' || ch == '_' || ch == '#' || ch == '@')
isSpecial = true;

if (ch >= 48 && ch <= 57)
isNum = true;

ch = _getch();
}

if (isSpecial && isNum)
cout << endl << "Password is valid" << endl;
else
cout << endl << "Password is not valid" << endl;
return 0;
}
```

Array of char - Palindrome

- Write a program that checks whether the input array of char is a palindrome string or not.

  The string is palindrome if its reverse is equal to the original string.

- Hints: - strcpy_s(destination, source) → copy the value from destination to source.

  - _strrev(array of char); → reverse the string itself in-place (no return value).

  - strcmp(destination, source) → compare twp strings and returns 0 if identical otherwise returns

<u>Sample of Execution:</u>

```
Enter string: abcba
Output: Palindrome

Enter string: abab
Output: Not Palindrome
```

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

int main()
{
char str1[10], str2[10];

cout << "Enter string: ";
cin >> str1;

strcpy_s(str2, str1);
_strrev(str2);
int compareVal = strcmp(str1, str2);

if(compareVal == 0)
cout << "String is Palindrome" << endl;
else
cout << "String is not Palindrome" << endl;

return 0;
}
```

String – Student Name

- Write a program that reads the first name and last name of a student and fill full name as first name concatenated with the first initial of the second name.

  Student is a struct with first name, last name and full name properties.

- Hints: - include library “string”

  - read string using getline(cin, s1).
  - s1.append(SecondString, StartIndexInSource, Count);
  - s1.insert(StartPosition, SubString);
  - s1.length();

Sample of Execution:

```
Enter first name: Mohamed
Enter Second name: Ahmed

Full name: Mohamed A.
```

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

struct Student {
string FirstName;
string SecondName;
string FullName;
};
```

```
int main()
{
Student std1;
cout << "Enter Firstname: ";
getline(cin, std1.FirstName);

cout << "Enter Secondname: ";
getline(cin, std1.SecondName);

std1.FullName = std1.FirstName.append(std1.SecondName, 0, 1);
std1.FullName.insert( std1.FullName.length() - 1, " ");
cout << "Full Name: " << std1.FullName << endl;

return 0;
}
```

Thank you!