Fundamentals of Computer Programming
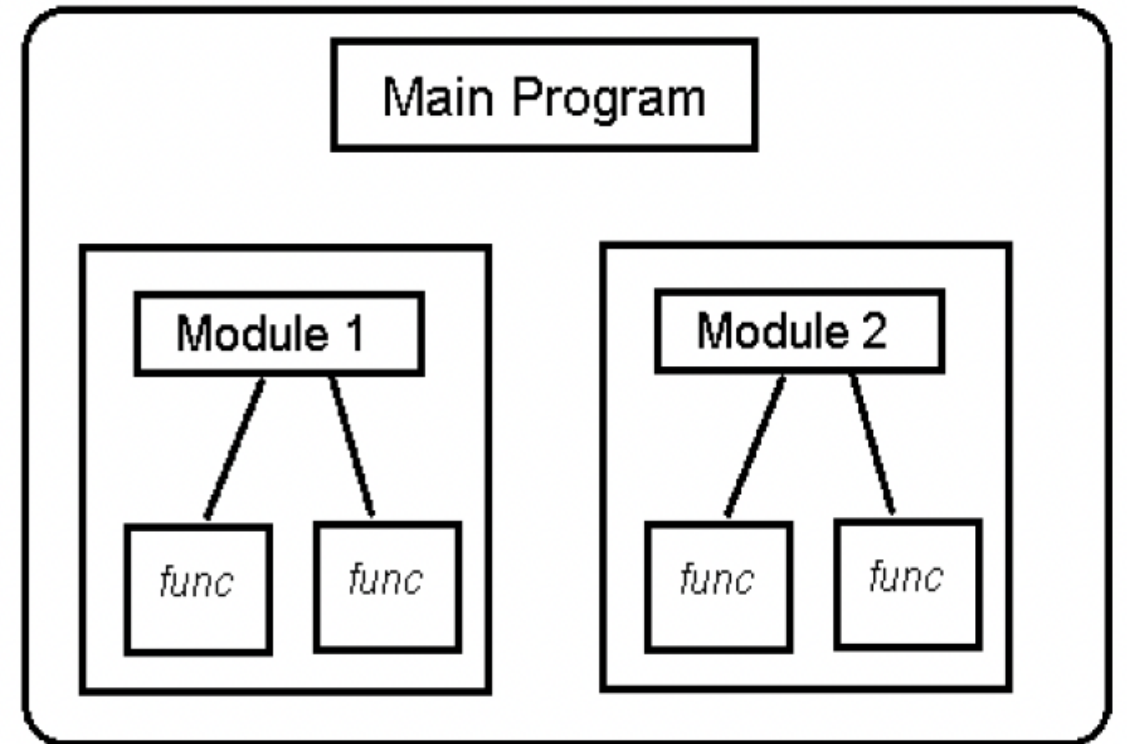
CS-110

Course Instructor: Dr. Momina Moetesum

Inbuilt Functions

Week 6-b

Learnning Objectives

01

To get familiarization with inbuilt functions and their libraries

02

To explore functions in iomanip.h

03

To explore functions in cmath.h

Inbuilt Functions

Functions	HeaderFiles
pow()	<cmath>
sqrt()	<cmath>
min()	<algorithm>
max()	<algorithm>
swap()	<utility>
gcd()	<numeric>
__gcd()	<algorithm>
toupper()	<cctype>/<ctype>
tolower()	<cctype>/<ctype>
floor()	<cmath>
ceil()	<cmath>

Pow()

- This function helps to find the value of a number raised to another number. It always takes to values of **double** data type as parameters (Also accepts int data type) and the result is of **double** data type.

OUTPUT: 5882.79

```cpp
// CPP program to illustrate power function

#include<iostream>
#include <cmath>
#include<iomanip>
using namespace std;
int main()
{
        double x = 6.1, y = 4.8;

        // Storing the answer in result.

        double result = pow(x, y);

        // Printing the result upto 2 decimal place

        cout<< fixed << setprecision(2) << result << endl;

        return 0;
}
```

Sqrt()

- This function helps to find the square root of any number. It takes **floating** pointer or **integer** data type as an argument. The result is returned after **rounding it** according to the required data type.

OUTPUT: 4.89898

```cpp
// CPP Program to demonstrate errors in double sqrt()
#include <cmath>
#include <iostream>
using namespace std;

// Driver Code
int main()
{
    int x = 24;
    double answer;

    answer = sqrt(x);

    // Printing square root of 24.
    cout << answer << endl;
    return 0;
}
```

Sqrt()

- This function helps to find the square root of any number. It takes **floating** pointer or **integer** data type as an argument. The result is returned after **rounding it** according to the required data type.

OUTPUT: 4.89898

```cpp
// CPP Program to demonstrate errors in double sqrt()
#include <cmath>
#include <iostream>
using namespace std;

// Driver Code
int main()
{
    int x = 24;
    double answer;

    answer = sqrt(x);

    // Printing square root of 24.
    cout << answer << endl;
    return 0;
}
```

Min()

- This function helps to find the minimum between two numbers. It takes two numbers of the **same** data **type** as arguments and returns the value of the minimum.

OUTPUT:5

```cpp
// C++ program to demonstrate the use of
std::min
#include <algorithm>
#include <iostream>
using namespace std;

int main()
{
    int a = 5;
    int b = 7;

    cout << std::min(a, b) << "\n";
    return 0;
}
```

Max()

- It helps in finding the maximum between two values. This function takes two values of the **same** data type as arguments and returns the value of the maximum element.

OUTPUT:
123
7

```cpp
// C++ program to demonstrate use of max()
#include <algorithm>
#include <iostream>
using namespace std;

int main()
{
    int a = 112, b = 123;

    // Comparing a and b
    cout << std::max(a, b) << "\n";

    // Returns the first one if both the numbers
    // are same
    cout << std::max(7, 7);
    return 0;
}
```

Swap()

- This function is used for swapping two numbers. It takes two values of the **same** data type as arguments and swaps their value.

Compiler Versions:
#include <algorithm> (until C++11)
#include <utility> (since C++11)
#include <string_view> (since C++17)

```cpp
// C++ program for illustration
// of swap() function
#include <iostream>
#include <utility>
using namespace std;

int main()
{
    int a = 10;
    int b = 20;
    cout << "Value of a before: " << a << endl;
    cout << "Value of b before: " << b << endl;

    // swap values of the variables
    swap(a, b);
    cout << "Value of a now: " << a << endl;
    cout << "Value of b now: " << b << endl;
    return 0;
}
```

Gcd()

- This function is used to find the GCD of two numbers. It takes two values of the **same data type as arguments** and returns the GCD of them.

#include <algorithm>
#include <numeric> (for C++17)

__gcd(value1, value2); [for C++14]
gcd(value1, value2);   [for C++17]

OUTPUT: gcd(6, 20) = 2

```cpp
// CPP program to illustrate
// gcd function of C++ STL
#include <algorithm>
#include <iostream>
#include<numeric> //for C++17

using namespace std;

int main()
{
    int a = 6, b = 20;
   // int ans = __gcd(a, b);
    int ans = gcd(a, b); //for C++17

    cout << "gcd(6, 20) = " << ans << endl;
    return 0;
}
```

Toupper()

- This function is used for converting a lowercase character to uppercase.

- int toupper(int ch)

OUTPUT:

m

M

```cpp
// C++ program to illustrate toupper()
method
#include <cctype>
#include <iostream>
using namespace std;

int main()
{
    char ch='m';
    cout<<ch<<endl;
    putchar(toupper(ch));
    return 0;
}
```

Tolower()

- This function is used for converting an uppercase character to lowercase.

OUTPUT:

M

m

```cpp
// C++ program to illustrate toupper()
method
#include <cctype>
#include <iostream>
using namespace std;

int main()
{
    char ch='M';
    cout<<ch<<endl;
    putchar(tolower(ch));
    return 0;
}
```

Floor()

- This function returns the largest possible integer value which is less than or equal to a given argument. It takes a floating number as an argument and returns an integer value.

OUTPUT:
Floor is: 2
Floor is: -3

```cpp
// C++ program to demonstrate floor function
#include <cmath>
#include <iostream>
using namespace std;

// Driver function
int main()
{
    // Using floor function which returns
    // floor of input value
    cout << "Floor is: " << floor(2.3) << "\n";
    cout << "Floor is: " << floor(-2.3) << "\n";

    return 0;
}
```

Ceil()

- This function is just the opposite of floor(), It returns the smallest possible integer value which is greater than or equal to the given argument. It takes a floating value as an argument and returns an integer value.

OUTPUT:
Ceil is: 3
Ceil is: -2

```cpp
// C++ program to demonstrate ceil
function
#include <cmath>
#include <iostream>
using namespace std;

// Driver function
int main()
{
    // Using ceil function which return
    // floor of input value
    cout << " Ceil is: " << ceil(2.3) << "\n";
    cout << " Ceil is: " << ceil(-2.3) << "\n";

    return 0;
}
```

Headerfile cmath.h/Arithmatic Functions

Function name and parameters	Parameters type	Function return value
acos(x)	x is a floating-point expression, $-1.0 \leq x \leq 1.0$	Arc cosine of x, a value between 0.0 and π
asin(x)	x is a floating-point expression, $-1.0 \leq x \leq 1.0$	Arc sine of x, a value between $-\pi/2$ and $\pi/2$
atan(x)	x is a floating-point expression	Arc tan of x, a value between $-\pi/2$ and $\pi/2$
ceil(x)	x is a floating-point expression	The smallest while number $\geq x$, ("ceiling" of x)
cos(x)	x is a floating-point expression, x is measured in radians	Trigonometric cosine of the angle
cosh(x)	x is a floating-point expression	Hyperbolic cosine of x
exp(x)	x is a floating-point expression	The value e raised to the power of x; (e = 2.718…)
fabs(x)	x is a floating-point expression	Absolute value of x
floor(x)	x is a floating-point expression	The largest whole number $\leq x$;

iomanip

Manipulator	Function
setw(int n)	To set the field width to n
setbase	To set the base of the number system
setprecision(int p)	The precision is fixed to p
setfill(char f)	To set the character to be filled
setiosflags(long l)	Format flag is set to l
resetiosflags(long l)	Removes the flags indicated by l
endl	Gives a new line
skipws	Omits white space in input
noskipws	Does not omit white space in the input
ends	Adds null character to close an output string
flush	Flushes the buffer stream
lock	Locks the file associated with the file handle
ws	Omits the leading white spaces present before the first field
hex, oct, dec	Displays the number in hexadecimal or octal or in decimal format

Further Exploration

- https://www.geeksforgeeks.org/precision-of-floating-point-numbers-in-c-floor-ceil-trunc-round-and-setprecision/

- •https://www.geeksforgeeks.org/ctype-hcctype-library-in-c-c-with-examples/

Acknowledgment

- Content of these slides are taken from:
    - https://www.geeksforgeeks.org/
    - https://www.tutorialspoint.com/
    - https://www.programiz.com/
    -