كليـــة العلـــــوم
قسم الأمن السيبراني

**Subject: Object Oriented Programming (OOP)**

**Second Stage**

**Lecturer: Dr. Abdulkadhem A. Abdulkadhem**

**Lecture (3)**

**Built-in Functions in C++**

**Built-in functions in C++** are predefined functions provided by the compiler, and they are part of standard libraries. These functions help developers perform various operations like working with numbers, strings, and memory management. The most common categories of built-in functions are:

1. Mathematical Functions
2. String Handling Functions
3. Input/Output Functions
4. Utility Functions

We'll discuss some of these functions and demonstrate their usage in code.

**1. Mathematical Functions in `<cmath>` Library**

**1.1 `ceil(x)` and `floor(x)`**

- `ceil(x)`: Returns the smallest integer greater than or equal to `x`.

- `floor(x)`: Returns the largest integer less than or equal to `x`.

 Example:

|  |
| --- |
| #include <iostream>#include <cmath> using namespace std;int main() { double num1 = 4.3; double num2 = 4.8; cout << "Ceil of " << num1 << " is: " << ceil(num1) << endl; // Output: 5 cout << "Floor of " << num2 << " is: " << floor(num2) << endl; // Output: 4 return 0;} |

1.2 `round(x)`

- Description: Returns the value of `x` rounded to the nearest integer.

 Example:

|  |
| --- |
| #include <iostream>#include <cmath>using namespace std;int main() { double num = 5.7; cout << "Round of " << num << " is: " << round(num) << endl; // Output: 6 return 0;} |

1.3 `sqrt(x)`

- Description: Calculates the square root of `x`.

Example:

|  |
| --- |
| #include <iostream>#include <cmath>using namespace std;int main() { double num = 16; cout << "Square root of " << num << " is: " << sqrt(num) << endl; // Output: 4 return 0;} |

1.4 `pow(x, y)`

- Description: Returns `x` raised to the power of `y` (i.e., \( x^y \)).

Example:

|  |
| --- |
| #include <iostream>#include <cmath>using namespace std;int main() { double base = 3; double exponent = 4; cout << base << " raised to the power of " << exponent << " is: " << pow(base, exponent) << endl; // Output: 81 return 0;} |

1.5 `abs(x)`

- Description: Returns the absolute value of `x`.

Example:

|  |
| --- |
| #include <iostream>#include <cmath>using namespace std;int main() { int num = -42; cout << "Absolute value of " << num << " is: " << abs(num) << endl; // Output: 42 return 0;} |

1.6 `sin(x)`, `cos(x)`, `tan(x)`

- Description: Compute the sine, cosine, and tangent of an angle `x` in radians.

Example:

|  |
| --- |
| #include <iostream>#include <cmath>using namespace std;int main() { double angle = 45.0; // Convert angle to radians double radian = angle \* M\_PI / 180.0; cout << "Sin(" << angle << ") = " << sin(radian) << endl; cout << "Cos(" << angle << ") = " << cos(radian) << endl; cout << "Tan(" << angle << ") = " << tan(radian) << endl; return 0;} |

1.7 `log(x)` and `log10(x)`

- `log(x)`: Returns the natural logarithm (base `e`) of `x`.

- `log10(x)`: Returns the base-10 logarithm of `x`.

Example:

|  |
| --- |
| #include <iostream>#include <cmath>using namespace std;int main() { double num = 100.0; cout << "Natural logarithm of " << num << " is: " << log(num) << endl; cout << "Base-10 logarithm of " << num << " is: " << log10(num) << endl; return 0;} |



**2. Random Number Functions in `<cstdlib>` Library**

**2.1 `rand()`:** Generates a pseudo-random number.

Example:

|  |
| --- |
| #include <iostream>#include <cstdlib>using namespace std;int main() { cout << "Random number: " << rand() << endl; // Output: A random number return 0;} |

**2.2 `srand(seed)` :** Seeds the random number generator with a value `seed`. Using different seeds ensures different sequences of random numbers.

Example:

|  |
| --- |
| #include <iostream>#include <cstdlib>#include <ctime>using namespace std;int main() { srand(time(0)); // Seed with current time cout << "Random number: " << rand() << endl; // Output: A random number return 0;} |

**3. Rounding Functions in `<cmath>` Library**

**3.1 `trunc(x)`:** Returns the integer part of `x`, removing the fractional part.

Example:

|  |
| --- |
| #include <iostream>#include <cmath>using namespace std;int main() { double num = 5.78; cout << "Truncated value of " << num << " is: " << **trunc(num)** << endl; // Output: 5 return 0;} |

**3.2 `fmod(x, y)`:** Returns the remainder of `x` divided by `y`.

Example:

|  |
| --- |
| #include <iostream>#include <cmath>using namespace std;int main() { double num1 = 7.0; double num2 = 3.0; cout << "Remainder of " << num1 << " / " << num2 << " is: " << fmod(num1, num2) << endl; // Output: 1.0 return 0;} |

**4. Utility Functions:**

**4.1 `exit(status)`:** Terminates the program immediately. The `status` indicates whether the program terminated normally (`0`) or with an error (any non-zero value).

Example:

|  |
| --- |
| #include <iostream>#include <cstdlib>using namespace std;int main() { cout << "Program will terminate now." << endl; exit(0); // Terminate with a normal status return 0; // This line will not be executed} |

4.2 **`system(command)`:**Executes a system command.

Example:

|  |
| --- |
| #include <iostream>#include <cstdlib>using namespace std;int main() { cout<<"hello world1"<<endl; // will be deleted cout<<"hello world2"<<endl; // also deleted cout<<"hello world3"<<endl; //deleted from secreen system("CLS"); // Clears the console (on Windows) cout << "Screen cleared." << endl; return 0;} |

Other using parameter

system("color b") 🡺 change the color.

system("date")🡺 deal with date.

**4.3 swap(a, b):** Swaps the values of a and b, in Library: <algorithm>

Example:

|  |
| --- |
| #include <iostream>#include <algorithm>using namespace std;int main() { int x = 10, y = 20; cout << "Before swap: x = " << x << ", y = " << y << endl; swap(x, y); cout << "After swap: x = " << x << ", y = " << y << endl; // Output: x = 20, y = 10 return 0;} |

4.4 max(a, b) and min(a, b): Returns the maximum or minimum of (a and b). Library: <algorithm>.

Example:

|  |
| --- |
| #include <iostream>#include <algorithm>using namespace std;int main() { int a = 15, b = 30; cout << "Maximum: " << max(a, b) << endl; // Output: 30 cout << "Minimum: " << min(a, b) << endl; // Output: 15 return 0;} |

**Summary**:

By using built-in functions, students can simplify their code and perform complex operations with minimal effort. The `<cmath>` library provides powerful mathematical functions like `sqrt()`, `pow()`, `ceil()`, and `floor()`, while `<cstdlib>` offers utilities for generating random numbers and interacting with the system. Understanding and utilizing these functions are crucial for solving mathematical problems efficiently in C++.

**Quiz: Built-in Functions in C++**

1. Which of the following headers is required to use the *sqrt* function?

1. <iostream>
2. <cmath>
3. <algorithm>
4. <stdlib.h>

2. What does the *ceil* function do?

1. Rounds a number down to the nearest integer
2. Rounds a number up to the nearest integer
3. Returns the absolute value of a number
4. Computes the square root of a number

3. What will *pow(2, 3)* return?

1. 2
2. 6
3. 8
4. 9

4. To round a number down to the nearest integer, you can use the \_\_\_\_\_\_ function.

5. The function \_\_\_\_\_\_ returns the absolute value of a number.