

Faculty of Computing

CS110: Fundamentals of Computer Programming

Class: BESE-16B

Lab 06: Introduction to Modular Programming

CLO 1	Understand the syntax and semantics of different programming constructs
CLO 2	Solve given real-world problem by applying appropriate programming concepts and techniques

Date: 14th September, 2025

Time: 2:00pm-5:00pm

Instructor: Dr. Momina Moetesum

Lab Engineer: Mr. Nadeem Nawaz



Introduction:

This lab introduces students with basics of modular programming in C++. Students will learn how to use in-built functions and will also learn to create their own functions. By working through guided exercises, students will practice applying these constructs to solve practical problems.

Learning Objectives:

After completing this section, you will be able to:

- **Apply** modular constructs to solve real-world problems by implementing C++ programs.
- **Understand** working of inbuilt functions using simple C++ programs and report outcomes.

Tools/Software Requirement:

- Microsoft Visual Studio (any version)
- C++ Compiler (integrated within Visual Studio)
- Word Processor (MS Word or equivalent for compiling deliverables)

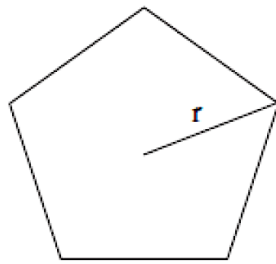
Task 1 [CLO 1]: Inbuilt Functions

Assume PI is 3.14159 and E is 2.71828, **print** output of the following function calls by writing them in a C++ program:

No.	Function Call	Output	No.	Function Call	Output
1	sqrt(4.0)		9	floor(-2.5)	
2	sin(2 * PI)		10	asin(0.5)	
3	cos(2 * PI)		11	acos(0.5)	
4	pow(2.0, 2)		12	atan(1.0)	
5	log(E)		13	ceil(-2.5)	
6	exp(1.0)		14	floor(2.5)	
7	max(2, min(3, 4))		15	log10(10.0)	
8	sqrt(125.0)		16	pow(2.0, 3)	

Task 2 [CLO 2]: Solving Geometrical Problems

Use the appropriate modular programming constructs to write a C++ program that prompts the user to enter the length from the center of a pentagon to a vertex and computes the area of the pentagon, as shown in the following figure.



$$Area = \frac{5 \times s^2}{4 \times \tan\left(\frac{\pi}{5}\right)}$$
$$s = 2r \sin\frac{\pi}{5}$$

The formula for computing the area of a pentagon is given above where s is the length of a side and can be computed by the formula also given above. r is the length from the center of a pentagon to a vertex. (*Note: Round up two digits after the decimal point.)

Sample test case:

Enter the length from the centre to a vertex: 5.5

The area of the pentagon is 71.92

Task 3 [CLO 1]: Formatting Output

Following are frequently used stream manipulators:

No.	Manipulators	Description
1	setprecision(n)	sets the precision of a floating-point number
2	fixed	displays floating-point numbers in fixed-point notation
3	showpoint	causes a floating-point number to be displayed with a decimal point with trailing zeros even if it has no fractional part
4	setw(width)	specifies the width of a print field
5	left	justifies the output to the left
6	right	justifies the output to the right
7	setprecision(n)	sets the precision of a floating-point number

Show the outputs of the following by writing the following lines of code in a C++ program including the required headerfiles:

```
double monthlyPayment = 1345.4567;
double totalPayment = 866.887234;
cout << setprecision(7);
cout << monthlyPayment << endl;
cout << totalPayment << endl;
cout << fixed << setprecision(2);
cout << setw(8) << monthlyPayment << endl;
cout << setw(8) << totalPayment << endl;
cout << left;
cout << setw(8) << monthlyPayment << endl;
cout << setw(8) << totalPayment << endl;
cout << right;
cout << setw(8) << monthlyPayment << endl;
cout << setw(8) << totalPayment << endl;
```



Task 4 [CLO 2]: Number Guessing Game

Use `rand()` function to write a C++ program that randomly generates an integer between 0 and 100, inclusive. The program prompts the user to enter a number. If the user's guess is correct, the program exits after displaying "Congratulations". However, in case of incorrect guess, the program tells the user whether the input is too low or too high, so the user can make the next guess intelligently. The user is given five chances to guess the number correctly. Here is a sample run:

```
Guess a magic number between 0 and 100

Enter your guess: 50 Enter
Your guess is too high

Enter your guess: 25 Enter
Your guess is too low

Enter your guess: 42 Enter
Your guess is too high

Enter your guess: 39 Enter
Yes, the number is 39
```

Deliverables:

Compile a single Word document as displayed in solution/answer part and submit this Word file on LMS.



Lab Rubrics:

Your Lab 6 will be graded out of 5 for each rubric according to the following rubrics.

Lab Rubrics for Lab 6					
Sr. No.	Assessment	Unacceptable (0 Marks)	Does Not Meet Expectations (1/2 Marks)	Meets Expectations (3/4 Marks)	Exceeds Expectations (5 Marks)
1	Illustrating the basic understanding of semantics and syntax (CLO1, PLO1)	The student did not submit any work. OR The student plagiarized the solution and/or used unfair means.	<p>The student is unable to demonstrate the understanding of syntax of C++ language and is unable to write an executable code.</p> <p>The student is not able to understand the structure of a program at all.</p>	<p>The student demonstrates some understanding of syntax of C++ language and is able to write a code with few errors.</p> <p>The student is able to understand the structure but still learning the syntax.</p>	<p>The student demonstrates good understanding of syntax of C++ language and is able to write executable code without help</p> <p>The student is able to understand the structure and is able to identify problems in the code when introduced.</p>
2	Application of Programming Concepts (CLO2, PLO3)		<p>The student is unable to apply the appropriate programming concepts to solve the given problem thus resulting in an incomplete or ineffective solution.</p> <p>The program flow is messy and incomprehensible. Codes are non-modular and cannot be reused.</p>	<p>The student requires some guidance to apply the appropriate programming concepts to solve the given problem.</p> <p>The program flow requires minor improvements. Codes are semi-modular and semi-reusable.</p>	<p>The student demonstrates a clear ability to apply the appropriate programming concepts to solve the given problem.</p> <p>The program flow is adequate. Codes are modular, reusable, and easily readable.</p>