UNIVERSITY OF SCHOLARS
Department of Computer Science and Engineering

Course Title:	**Structured Programming Language Lab**
Course Code:	CSE 1214
Credits:	1.0
Prerequisites:	N/A
Course Instructor:	**Md. Nurul Islam**
Contact Number:	+8801521470560
Email:	**nisakib@ius.edu.bd**

Course Description:
This course focuses on the syntax, semantics of structured programming while analyzing and designing various applications using different library functions. Also, it helps to develop basic programming and problem-solving skills to program design and development.

Objectives:
To understand the basics of structured programming, keywords and syntax. To understand typical characteristics, mechanisms and solve problems using structured programming language. To develop basic programming skills with respect to program design and development.

Grading policy:
Attendance: 10%
*Class Performance and class tests: 20%
Mid-Term: 30%
Final: 40%
*Class performance includes a series of class tests, assignments, and class performance.

Programming Language: C

Software: CodeBlocks

What is Program:

A program is a set of instructions that a computer uses to perform a specific function.

1. Introduction to CodeBlocks

What is codeBlocks:

CodeBlocks is an open-source, cross-platform (Windows, Linux, MacOS), and free C/C++ IDE. It supports many compilers, such as GNU GCC (MinGW and Cygwin) and MS Visual C++. It supports interactive debugging (via GNU GDB or MS CDB).

Step 1: Download

Goto http://www.codeblocks.org/downloads. Click "Download the binary release". Select your operating platform (e.g., Windows 2000/XP/Vista/7/10/11). Download the installer with GCC Compiler, e.g., codeblocks-20.03mingw-setup.exe (145.35 MB) (which includes MinGW's GNU GCC compiler and GNU GDB debugger).

Step 2: Install

Run the downloaded installer. Accept the default options.

Verify the Compiler's and Debugger's Path: (For CodeBlocks 13.12 For Windows) Goto "Settings" menu ⇒ "Compiler..." ⇒ In "Selected Compiler", choose "GNU GCC Compiler" ⇒ Select tab "Toolchain Executables" ⇒ Check the "Compiler's Installation Directory". It shall be set to the "MinGW" sub-directory of the CodeBlocks installation directory, for example, suppose that CodeBlocks is installed in "c:\Program Files\codeblocks", set it to "c:\Program Files\codeblocks\MinGW".

Similarly, check the debugger's path. Goto "Settings" menu ⇒ "Debugger..." ⇒ Expand "GDB/CDB debugger" ⇒ Select "Default" ⇒ In "Executable path", provide the full-path name of "gdb.exe", for example, "c:\Program Files\codeblocks\MinGW\bin\gdb.exe".

Step 3: How to open codeblocks and save any program

- Go to search option in your Desktop/ Laptop from taskbar
- Search codeBlocks
- Click codeBlocks icon
- Now goto **File -> New -> Empty file** from top left side in your window
- Or just press **Ctrl+Shift+N**

- To save your program
- Goto **File-> Save as ->** select your local drive "or just press Ctrl+s"
- Give program name and program extension ".c"
- And press the save button

2. Syntax rules in C programming

1. C is a case sensitive language so all C instructions must be written in lower case letter.
2. All C statements must end with a semicolon.
3. Whitespace is used in C to describe blanks and tabs.
4. Whitespace is required between keywords and identifiers

3. Printf function (string print and string print with new line)

a. **Write a program that prints "Hello World"**.

Solve:

#include <stdio.h>

int main() {

   printf("Hello World");

   return 0;

}

b. **Write a program that prints Your name.**

 Solve:

#include <stdio.h>

int main() {

   printf("ABC");

   return 0;

}

c. **Write a program that prints Your name, Your Id, Your University name. Every information should be printed in separate lines.**

Solve:

#include <stdio.h>

int main() {

   printf("Name: \nId: \nUniversity name: ");

 return 0;

 }

4. Comments

There are two ways in which we can write comments.

a. **Using //: This is used to write a single-line comment.**

Write a program that prints Your name, Your Id, Your University name. Every information should be printed in separate lines.(Use comments)

Solve:

#include <stdio.h>

int main(){

   printf("Name: \nId: \nUniversity name: ");   //Printing Name, Id, University Name

   return 0;

 }

b. **Using /* */: Anything enclosed within /* and */ , will be treated as multi-line comments.**

/*

   This is my first program.

   I am very excited!

```
*/

#include <stdio.h>

int main()

{

   // Printing Hello World

   printf("Hello,World");

   return 0;

}
```

5. Data types & Variable initialization

a. Write a program to declare an integer variable.

Solve:

```
#include <stdio.h>

int main() {

int myNum = 15;        // Integer (whole number)

 return 0;

}
```

b. Write a program to declare a float variable.

Solve:

```
#include <stdio.h>

int main() {

float myNum = 15.55;// Float (fraction number)

 return 0;

}
```

c. **Write a program to declare a character variable.**

Solve:

#include <stdio.h>

int main() {

char myChar = 'A'; // Character

 return 0;

}

d. **Write a program to declare a character variable. (Alternative way)**

Solve:

#include <stdio.h>

int main() {

char myChar, myChar1;

 myChar= 'A'; // Character

myChar1= 'B'; // Character

 return 0;

}

6. Format specifier

a. **Write a program that prints an integer variable. (Initialize and Print)**

Solve:

#include <stdio.h>

int main() {

int myNum = 15;   // Integer (whole number)

printf("%d",myNum);

 return 0;

}

b. **Write a program that prints a float variable.**

   Solve:

   #include <stdio.h>

   int main() {

   float myNum = 15.55;// Float (fraction number)

   printf("%f",myNum);

    return 0;

   }

c. **Write a program to declare a character variable.**

   Solve:

   #include <stdio.h>

   int main() {

   char myChar = 'A'; // Character

   printf("%c",myChar);

    return 0;

   }

   Sample:

   int myNum = 5;

   float myFloatNum = 5.99;   // Floating point number
   char myLetter = 'D';   // Character

```
// Print variables
printf("%d\n", myNum);
printf("%f\n", myFloatNum);
printf("%c\n", myLetter);
```

----

d. Write a program that prints an integer variable in a sentence. (Initialize and Print)

Solve:

```
#include <stdio.h>

int main() {

int myNum = 15;      // Integer (whole number)

printf("The number is : %d",myNum );

 return 0;

}
```

e. Write a program to declare and print two character variables. (Alternative way)

Solve:

```
#include <stdio.h>

int main() {

char myChar, myChar1;

 myChar= 'A'; //  Character

myChar1= 'B'; //  Character

printf("%c %c",myChar,myChar1);

 return 0;

}
```

7. (User input and output)

a. **Write a program to take an integer value as input from the user.**

Solve:

```c
#include <stdio.h>
int main() {
    int num;
    printf("Enter any integer value: ");
    scanf("%d",&num);
    printf("%d",num);
    return 0;
}
```

b. **Write a program to take a float value as input from the user.**

Solve:

```c
#include <stdio.h>
int main() {
    float num;
    printf("Enter any float value: ");
    scanf("%f",&num);
    printf("%f",num);
    return 0;
}
```

c. **Write a program to take a character value input from user**

Solve:

```c
#include <stdio.h>
int main() {
```

```c
    char variable;

    printf("Enter a Character: ");

    scanf("%c",&variable);

    printf("%c",variable);

    return 0;

}
```

d. **Write a program to take a double value as input from the user.**

   Solve:

```c
#include <stdio.h>

int main() {

    double num;

    printf("Enter a Double value: ");

    scanf("%lf",&num);

    printf("%lf",num);

    return 0;

}
```

e. **Write a program to print integer value**

   Solve:

```c
#include <stdio.h>

int main() {

    int num = 14;

    printf("The value of integer num: ");

    printf("%d",num);
```

```
    return 0;

}
```

f.   **Write a program to print float value**

Solve:

```
#include <stdio.h>
int main() {
    float num = 2.33;
    printf("The value of float num: ");
    printf("%f",num);
    return 0;
}
```

g. **Write a program to print character variable**

Solve:

```
#include <stdio.h>
int main() {
    char var = 'A';
    printf("Your character is: ");
    printf("%c", var);
    return 0;
}
```

h. **Write a program to print double value**

Solve:

```c
#include <stdio.h>

int main() {

    double num = 100.3345;

    printf("The value of double num: ");

    printf("%lf",num);

    return 0;

}
```

8. Constant :

a.Write a program to declare an integer (int) as a constant variable.

Solve:

```c
#include <stdio.h>

int main() {

    const int myNum = 15 ; // myNum will always be 15

    myNum = 10; //error: assignment of read-only variable 'myNum'

    printf("The value of number: ");

    printf("%d",num);

    return 0;

}
```

9. Operator (Arithmetic operator, Unary operator, Assignment operator, Comparison operator, Logical operator, sizeof operator)

Arithmetic Operators:

Operator	Name	Description	Example

+	Addition	Adds together two values	x + y
-	Subtraction	Subtracts one value from another	x - y
*	Multiplication	Multiplies two values	x * y
/	Division	Divides one value by another	x / y
%	Modulus	Returns the division remainder	x % y
++	Increment	Increases the value of a variable by 1	x++
--	Decrement	Decreases the value of a variable by 1	x--

a. **Write a program that calculates and displays the sum of two integer numbers.**

   Solve:

   #include <stdio.h>
   int main() {

```c
    int num1,num2,sum;
    num1=10;
    num2=5;
    sum = num1+num2;
    printf("The sum is : ");
    printf("%d",sum); // output will be 15
    return 0;
}
```

b. **Write a program that calculates and displays the difference between two integer numbers.**

   Solve:

```c
#include <stdio.h>
int main() {
    int num1,num2,diff;
    num1=10;
    num2=5;
    diff = num1-num2;
    printf("The difference is : ");
    printf("%d",diff); // output will be 5
    return 0;
}
```

c. **Write a program that calculates and displays the multiplication of two integer numbers.**

   Solve:

```c
#include <stdio.h>
int main() {
    int num1,num2,multiplication;
    num1=10;
```

```c
    num2=5;
    multiplication = num1*num2;
    printf("The multiplication is : ");
    printf("%d",multiplication); // output will be 50
    return 0;
}
```

d. **Write a program that calculates and displays the division of two integer numbers.**

Solve:

```c
#include <stdio.h>
int main() {
    int num1,num2,div;
    num1=10;
    num2=5;
    div= num1/num2;
    printf("The result is : ");
    printf("%d",div); // output will be 50
    return 0;
}
```

e. **Write a program that calculates and displays the division of two float numbers.**

Solve:

```c
#include <stdio.h>
int main() {
    float num1,num2,div;
    num1=13;
    num2=2;
```

```c
    div= num1/num2;
    printf("The result is : ");
    printf("%f",div); // output will be 50
    return 0;
}
```

f. **Write a program that takes two integer numbers from the user and displays the sum of the numbers.**

Solve:

```c
#include <stdio.h>
int main() {
    int num1,num2,sum;
    printf("Enter the first number: ");
    scanf("%d",&num1);
    printf("\nEnter the second number: ");
    scanf("%d",&num2);
    sum = num1+num2;
    printf("%d",sum);
    return 0;
}
```

g. **Write a program that takes two integer numbers from the user. Now, increase those numbers with the Arithmetic operator and print the result.**

Solve:

```c
#include <stdio.h>
int main() {
    int num1,num2,sum;
    printf("Enter the first number: ");
    scanf("%d",&num1);
```

```c
printf("\nEnter the second number: ");
scanf("%d",&num2);
num1++;
num2++;
printf("First number : %d \nSecond Number: %d",num1,num2);
return 0;
}
```

h. **Write a program that takes two integer numbers from the user. Now, decrease those numbers with the Arithmetic operator and print the result.**

Solve:

```c
#include <stdio.h>
int main() {
    int num1,num2,sum;
    printf("Enter the first number: ");
    scanf("%d",&num1);
    printf("\nEnter the second number: ");
    scanf("%d",&num2);
    num1--;
    num2--;
    printf("First number : %d \nSecond Number: %d",num1,num2);
    return 0;
}
```

Assignment Operators:

Operator	Example	Same As
=	x = 5	x = 5
+=	x += 3	x = x + 3

-=	x -= 3	x = x - 3
*=	x *= 3	x = x * 3
/=	x /= 3	x = x / 3
%=	x %= 3	x = x % 3
&=	x &= 3	x = x & 3
|=	x |= 3	x = x | 3
^=	x ^= 3	x = x ^ 3
>>=	x >>= 3	x = x >> 3
<<=	x <<= 3	x = x << 3

i. **Write a program that takes an integer number from the user. Now, add 3 with the number using Assignment Operators and display the result.**

Solve:

```
#include <stdio.h>

int main() {
        int x;
        printf("Enter the number : ", x);
        scanf("%d",&x);
```

```
        x += 3;
        printf("%d", x);
         return 0;
  }
```

j. **Write a program that takes an integer number from the user. Now, subtract 3 with the number using Assignment Operators and display the result.**

Solve:

```
#include <stdio.h>

int main() {
         int x;
        printf("Enter the number : ", x);
        scanf("%d",&x);
        x -= 3;
        printf("%d", x);
         return 0;
  }
```

Comparison Operators :

Operator	Name	Example

==	Equal to	x == y	
!=	Not equal	x != y	
>	Greater than	x > y	
<	Less than	x < y	
>=	Greater than or equal to	x >= y	
<=	Less than or equal to	x <= y	

Logical Operators:

Operator	Name	Description	Example

&&	Logical and	Returns true if both statements are true	x < 5 && x < 10
||	Logical or	Returns true if one of the statements is true	x < 5 || x < 4
!	Logical not	Reverse the result, returns false if the result is true	!(x < 5 && x < 10)

10. If/ else statement

- The if-else statement executes a block of code if a specific condition is true. If the condition is false then it executes another block of code.

a. **Write a program to find out the small number between two numbers. (Using only 'IF')**

Solve:

```
#include <stdio.h>
int main()
{
    int x = 20;
    int y = 22;
    if (x<y)
    {
        printf("Variable x is less than y");
    }
    return 0;
}
```

b. **Write a program to check given number is zero or not**

Solve:

```c
#include<stdio.h>
int main()
{
int no;

printf("Enter the number\n");
scanf("%d",&no);

if(no==0)
{
 printf("Given number is zero");
}
else
{
 printf("Given number is not zero");
}
return 0;
}
```

c. **Write a program To Print Given Number Is Even or Odd**

Solve:

```c
#include<stdio.h>

void main()

{

int no;

printf("Enter the number\n");

scanf("%d",&no);

if(no%2==0)

printf("Given number is Even\n");

else
```

```
    printf("Given number is Odd\n");

}
```

d. **Write a program that reads two numbers from the user and displays if they are equal or greater than or less than from each other.**

Solve:

```
#include <stdio.h>
int main()
{
    int x, y;
    printf("enter the value of x:");
    scanf("%d", &x);
    printf("enter the value of y:");
    scanf("%d", &y);
    if (x>y)
    {
        printf("x is greater than y\n");
    }
    else if (x<y)
    {
        printf("x is less than y\n");
    }
    else if (x==y)
    {
        printf("x is equal to y\n");
    }
    printf("End of Program");
    return 0;
}
```

e. **Write a program to check whether a character is alphabet or not**

Solve:

```c
#include <stdio.h>

int main()
{
    char ch;

    printf("Enter any character: ");
    scanf("%c", &ch);


    if((ch >= 'a' && ch <= 'z') || (ch >= 'A' && ch <= 'Z'))
    {
        printf("Character is an ALPHABET.");
    }
    else
    {
        printf("Character is NOT ALPHABET.");
    }

    return 0;
}
```

f. Write a program to check whether a character is alphabet, digit or special character

Solve:

```c
#include <stdio.h>

int main()
{
    char ch;

    printf("Enter any character: ");
    scanf("%c", &ch);


    if((ch >= 'a' && ch <= 'z') || (ch >= 'A' && ch <= 'Z'))
    {
        printf("'%c' is alphabet.", ch);
    }
```

```c
        else if(ch >= '0' && ch <= '9')
        {
            printf("'%c' is digit.", ch);
        }
        else
        {
            printf("'%c' is special character.", ch);
        }

        return 0;
    }
```

g. Write a program to check whether a character is Uppercase or Lowercase

Solve:

```c
#include <stdio.h>

int main()
{
    char ch;
    printf("Enter any character: ");
    scanf("%c", &ch);


    if(ch >= 'A' && ch <= 'Z')
    {
        printf("'%c' is uppercase alphabet.", ch);
    }
    else if(ch >= 'a' && ch <= 'z')
    {
        printf("'%c' is lowercase alphabet.", ch);
    }
    else
    {
        printf("'%c' is not an alphabet.", ch);
    }

    return 0;
}
```