

Experiment No. 2

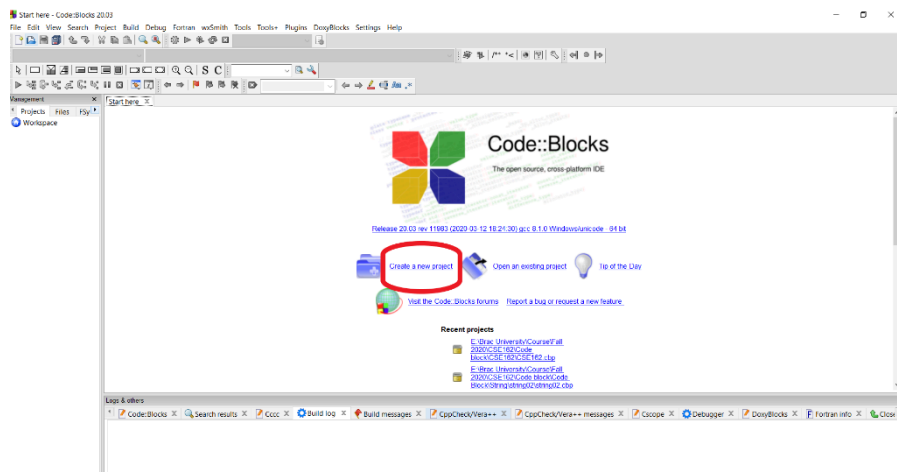
Introduction to C programming, I/O, and Datatypes

1. **Objective:** The Lesson is designed to teach the fundamental programming concepts and methodologies in the C programming language via laboratory experiences. CodeBlocks is the Integrated Development Environment (IDE) that will be used.
2. **Learning Outcome:** Upon completion of this experiment, the students shall be able to:
 - Write C programs on CodeBlocks
 - Perform input/output(I/O) operations using C programming language.
 - Define data types and use them in simple data processing applications
3. **Software requirements:**
 - Any IDE that supports a C compiler (preferably Code::Blocks; any other IDE or text editor with a C compiler installed will also be accepted.)
4. **Example Problems**

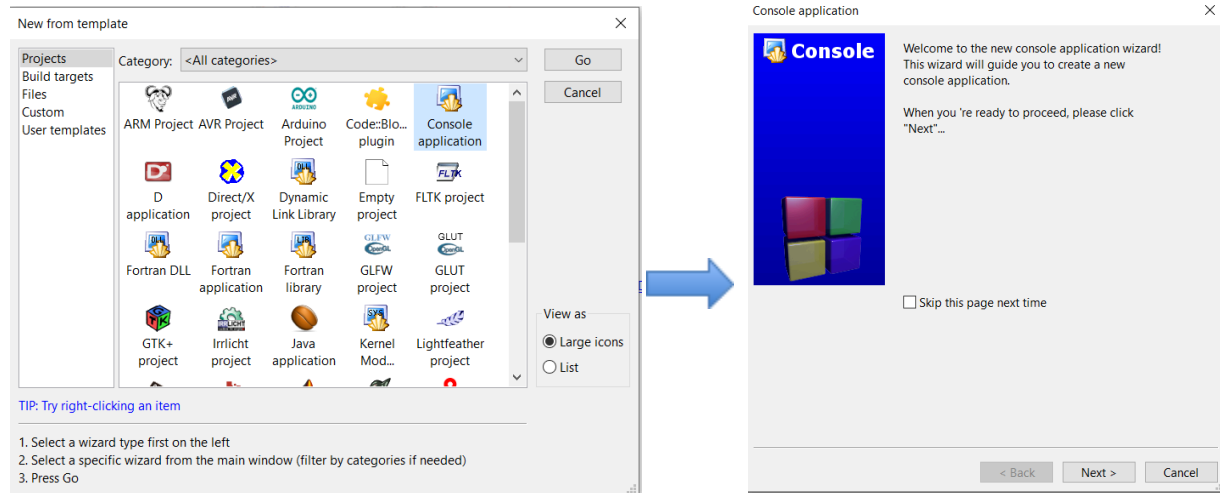
Example 1: Writing C programs in CodeBlocks

To start a new project click on

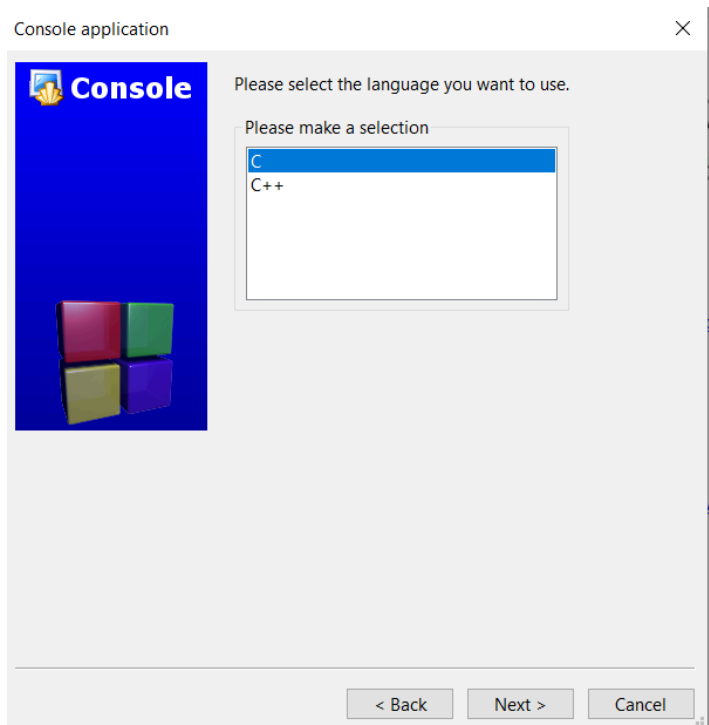
File >New >Project Or click “Create a new project”



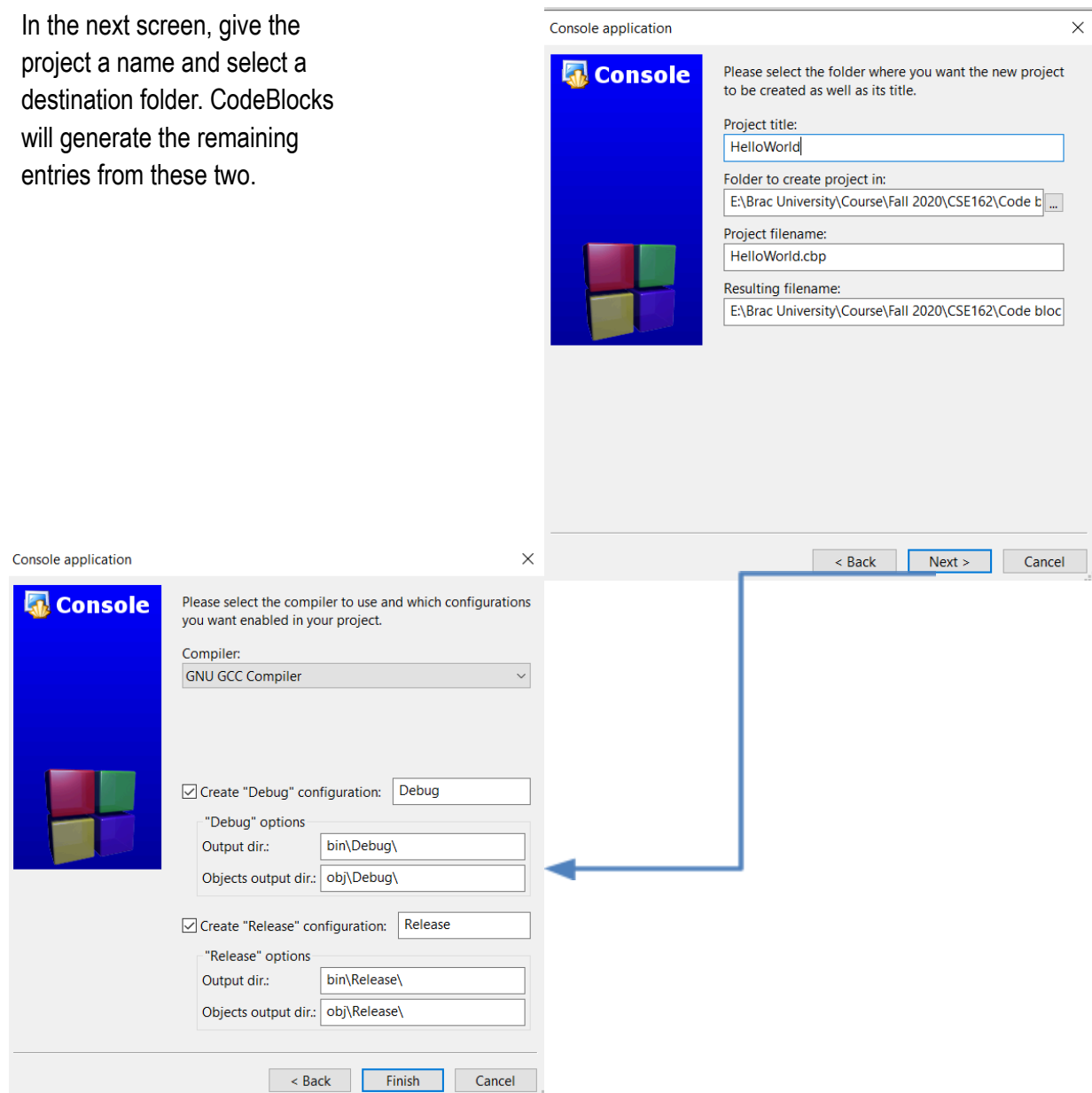
Select Console application, as this is the most common for general purposes, and click Go.



Continue through the menus, select C when prompted for a language

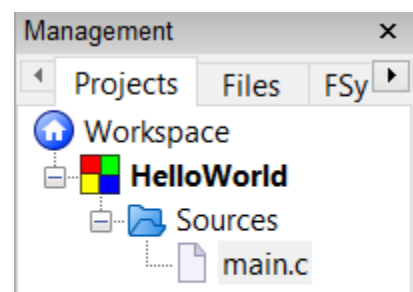


In the next screen, give the project a name and select a destination folder. CodeBlocks will generate the remaining entries from these two.



Press finish and the project will be generated.

The main window will turn gray, the source file needs only to be opened. In the Projects tab of the Management pane



on the left expand the folders and double click on the source file main.c to open it in the editor.

Code:

```
#include <stdio.h>
#include <stdlib.h>

//How to add comments- This is a way
/*This is my first lab, so I create
a test file - this is another way*/

int main()
{
    printf("Hello world!\n");

    //this is "\n" for creating a new line

    //this is "\t" for creating a tab

    //To print the value of a variable to the screen,

    //we will use the printf() function.

    //we will at first print I like Programming

    printf ("I like programming");

    return 0;
}
```

Example 3: Use of Conversion Specifiers

Code:

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int try_integer = -67;
    unsigned int try_ui = 33;
    float try_float = 9.82;
    double try_double = 573717.1817818;
    char try_char = 'y';
    char try_string[] = "Let's try to print it";

    printf ("Integer: %d\n", try_integer);
    printf ("Unsigned integer: %u\n", try_ui);

    printf ("Floating-point: %f\n", try_float);
    printf ("Double, exponential notation: %17.11e\n", try_double);

    printf ("Single character: %c\n", try_char);
    printf ("String: %s\n", try_string);

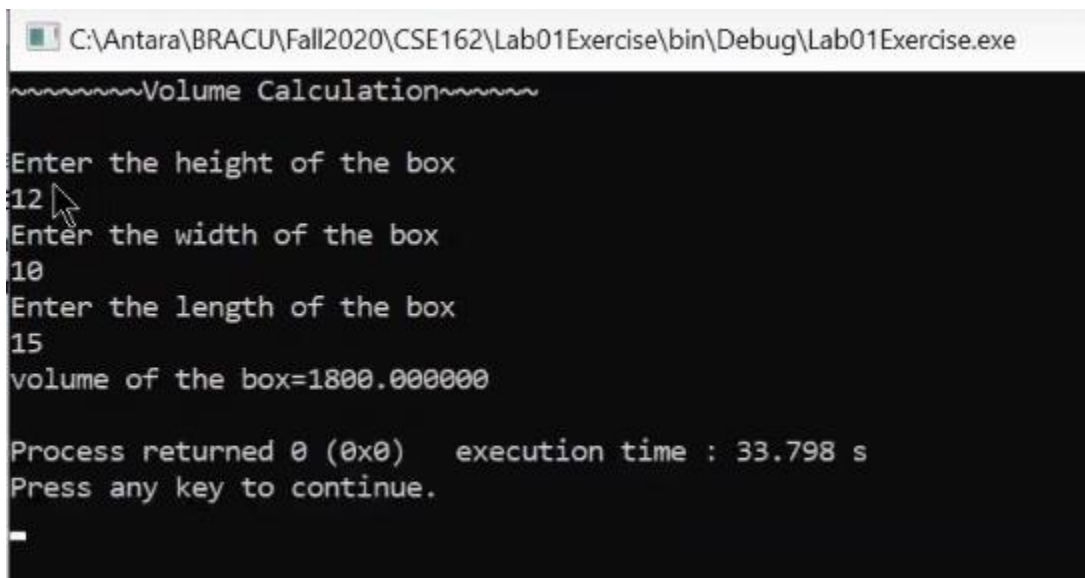
    printf ("%s received the %s award this year\n", "John", "Leadership" );
    printf ("I took %d courses in this semester\n", 4);
    printf ("The pi value is %f\n", 3.1415924653);
    printf ("The pi value is %.2f", 3.1415924653);

    return 0;
}
```

Example 4: Write a program in C to calculate and display the volume of a box based on the user inputs length, width and height.

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main()
5  {
6      printf("~~~~~Volume Calculation~~~~~\n\n");
7      int h;
8      int w, l;
9      float vol;
10     printf("Enter the height of the box\n"); scanf("%d", &h);
11     printf("Enter the width of the box\n"); scanf("%d", &w);
12     printf("Enter the length of the box\n"); scanf("%d", &l);
13     vol=h*w*l;
14     printf("volume of the box=%f\n", vol);
15     return 0;
16 }
17
```

Output:



```
C:\Antara\BRACU\Fall2020\CSE162\Lab01Exercise\bin\Debug\Lab01Exercise.exe
~~~~~Volume Calculation~~~~~
Enter the height of the box
12
Enter the width of the box
10
Enter the length of the box
15
volume of the box=1800.000000

Process returned 0 (0x0)   execution time : 33.798 s
Press any key to continue.
_
```

5. Results after running the example programs:

Screenshots of the results:

6. Lab report assignments (submit source code and screenshots of results):

Task 1: First find the value of each of the following arithmetic expressions in C. Then step by step, show in detail, how to hand calculate those values? Simply, you will write your investigation report on how C in CodeBlocks found those answers.

For example, if you are told to write $5+4*3/2-1$, Your answer should be like:

$= 5+4*3/2-1 = 5+12/2-1 = 5+6-1 = 11-1 = 10$, in the investigation report

- A) $3+4.8*2$
- B) $12\%16$
- C) $2.*6/5$
- D) $5.2+12/8$
- E) $4-(6+18.0/3)/3$
- F) $17-12*4$
- G) $2*6/5.$
- H) $7/(4*2)$
- I) $2*6/5$
- J) $19\%13$
- K) $36/60$

Task 2: Try the following programs and note what happens. Then try to explain the behaviors line by line.

```
int main()
{
    char b;
    int i = 257;
    double d = 323.142;
    b = i;
    printf("%c\n",b);
    b = (char) i;
    printf("%c\n",b);
    i = (int) d;
    printf("%d\n",i);
    b = (char) d;
    printf("%c\n",b);
}
```

Task 3: Here is a pseudo program that computes the number of miles that light will travel in a specified number of days. Try to compile and run the following C program.

If it does not work, make necessary corrections so that it compiles, runs and gives correct output.

```
Lightspeed = 186000
days = 1000;          //specify number of days here
seconds = Days*24*60*60; /*convert to seconds/
distance = lightspeed * Second;    //compute distance
printf("In " + days + " days light will travel about ");
printf(Distance + "miles.");
```

Task 4) Write a program in C that can calculate the area of a circle up to 4 decimal points taking the radius as input. Take the value of radius from the user in the console window and radius can be any integer or decimal fraction number. Print the value of the area.

(Submit your code in report by copy pasting it and also attach a screenshot of your result in command window)

7. Comment/Discussion on the obtained results and discrepancies (if any).