Course Code: CL-1004	Course: Object Oriented Programming Lab

Spring 2024, Lab Manual - 01

Contents

INTRODUCTION TO VISUAL STUDIO CODE 2023

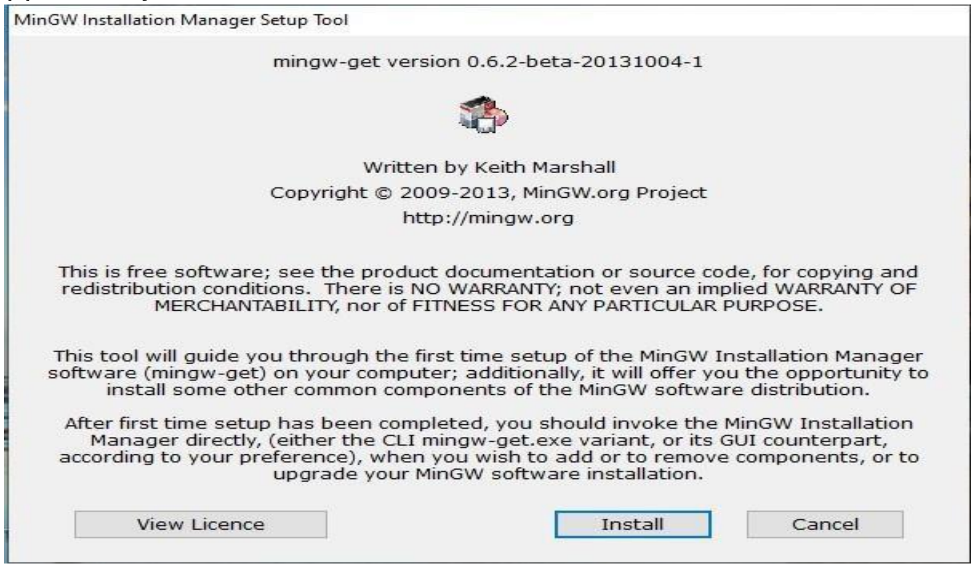
1. First, you have to download and install the Visual Studio. For that, you can refer to https://code.visualstudio.com/.
2. After you installed the IDE, you need to configure the VS Code for the C++ compiler.

MinGW FOR C++ COMPILER

MinGW software consists of three basic components: the assembler, linker, and compiler. It is sufficient to begin designing applications with this minimal tool set.

MinGW C++ Download and Installation

1. First you have to download and install the MinGW for the C++ compiler. For that, you can refer to https://sourceforge.net/projects/mingw/
2. After you download the MinGW, Run the setup and then following window will appear on your screen.


MinGW Installation Manager Setup Tool

mingw-get version 0.6.2-beta-20131004-1

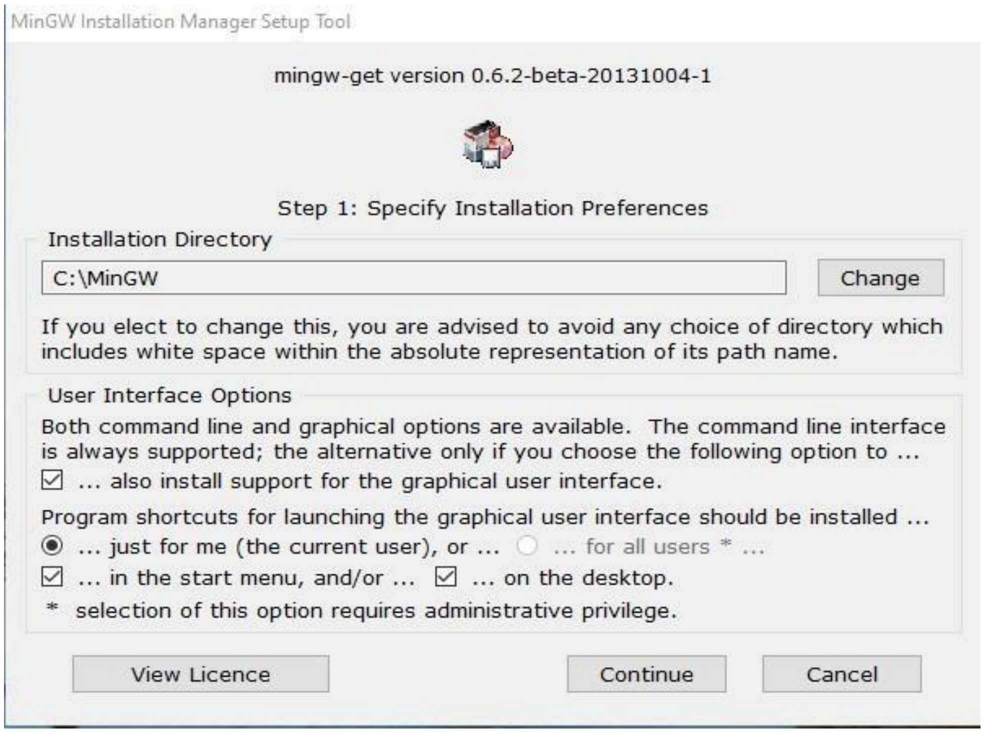
Written by Keith Marshall
Copyright © 2009-2013, MinGW.org Project
http://mingw.org

This is free software; see the product documentation or source code, for copying and redistribution conditions. There is NO WARRANTY; not even an implied WARRANTY OF MERCHANTABILITY, nor of FITNESS FOR ANY PARTICULAR PURPOSE.

This tool will guide you through the first time setup of the MinGW Installation Manager software (mingw-get) on your computer; additionally, it will offer you the opportunity to install some other common components of the MinGW software distribution.

After first time setup has been completed, you should invoke the MinGW Installation Manager directly, (either the CLI mingw-get.exe variant, or its GUI counterpart, according to your preference), when you wish to add or to remove components, or to upgrade your MinGW software installation.

3. Click **Install**.
The following pop-up window will appear.


MinGW Installation Manager Setup Tool

mingw-get version 0.6.2-beta-20131004-1

Step 1: Specify Installation Preferences

Installation Directory

C:\MinGW [Change]

If you elect to change this, you are advised to avoid any choice of directory which includes white space within the absolute representation of its path name.

User Interface Options

Both command line and graphical options are available. The command line interface is always supported; the alternative only if you choose the following option to ...

☑ ... also install support for the graphical user interface.

Program shortcuts for launching the graphical user interface should be installed ...

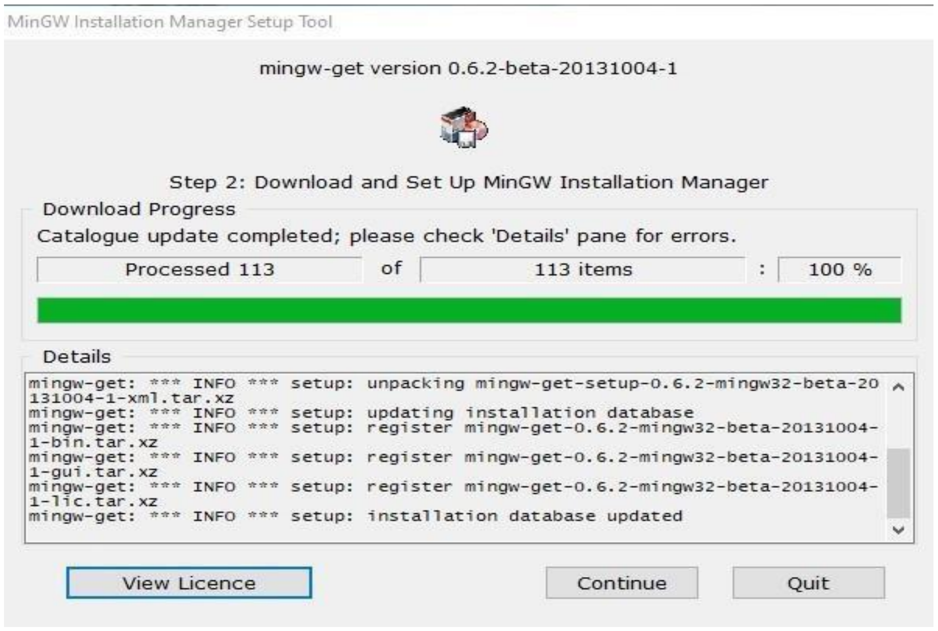
● ... just for me (the current user), or ... ○ ... for all users * ...

☑ ... in the start menu, and/or ... ☑ ... on the desktop.

* selection of this option requires administrative privilege.

You can install this software anywhere, but I recommend installing it in the default directory: **C:\MinGW**.

4. Click **Continue.** The following pop-up window will appear, showing the downloading progress. After about a minute, it should appear as follows.


MinGW Installation Manager Setup Tool

mingw-get version 0.6.2-beta-20131004-1

Step 2: Download and Set Up MinGW Installation Manager

Download Progress

Catalogue update completed; please check 'Details' pane for errors.

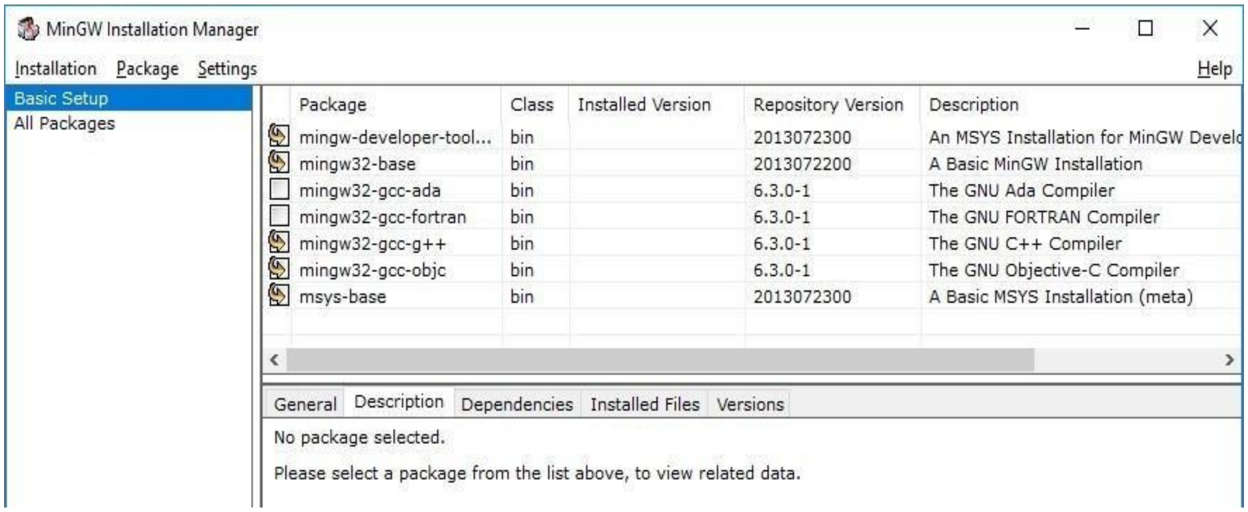
Processed 113 of 113 items : 100 %

Details

```
mingw-get: *** INFO *** setup: unpacking mingw-get-setup-0.6.2-mingw32-beta-20
131004-1-xml.tar.xz
mingw-get: *** INFO *** setup: updating installation database
mingw-get: *** INFO *** setup: register mingw-get-0.6.2-mingw32-beta-20131004-
1-bin.tar.xz
mingw-get: *** INFO *** setup: register mingw-get-0.6.2-mingw32-beta-20131004-
1-gui.tar.xz
mingw-get: *** INFO *** setup: register mingw-get-0.6.2-mingw32-beta-20131004-
1-lic.tar.xz
mingw-get: *** INFO *** setup: installation database updated
```

5. Click **Continue**.

   The following pop-up window will appear. Ensure on the left that Basic Setup is highlighted. Click the five boxes indicated below: mingw- developer-tool, mingw32- base, mingw32-gcc-g++, mingw32-gcc-obj, msys-base. After clicking each, select Mark for selection. This window should appear as follows.

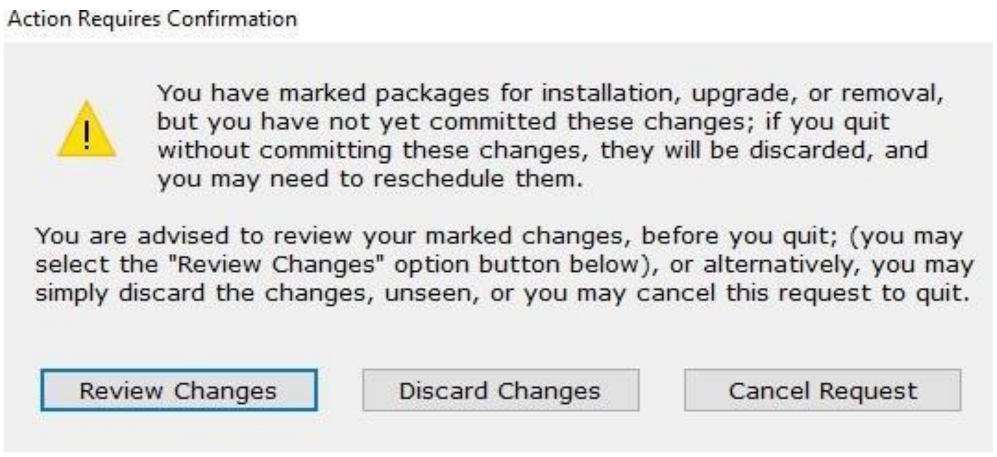


Package	Class	Installed Version	Repository Version	Description
mingw-developer-tool...	bin		2013072300	An MSYS Installation for MinGW Develo
mingw32-base	bin		2013072200	A Basic MinGW Installation
mingw32-gcc-ada	bin		6.3.0-1	The GNU Ada Compiler
mingw32-gcc-fortran	bin		6.3.0-1	The GNU FORTRAN Compiler
mingw32-gcc-g++	bin		6.3.0-1	The GNU C++ Compiler
mingw32-gcc-objc	bin		6.3.0-1	The GNU Objective-C Compiler
msys-base	bin		2013072300	A Basic MSYS Installation (meta)

No package selected.

Please select a package from the list above, to view related data.

6. Terminate (click **X** on) the **MinGW Installation Manager**. The following pop- up window should appear.

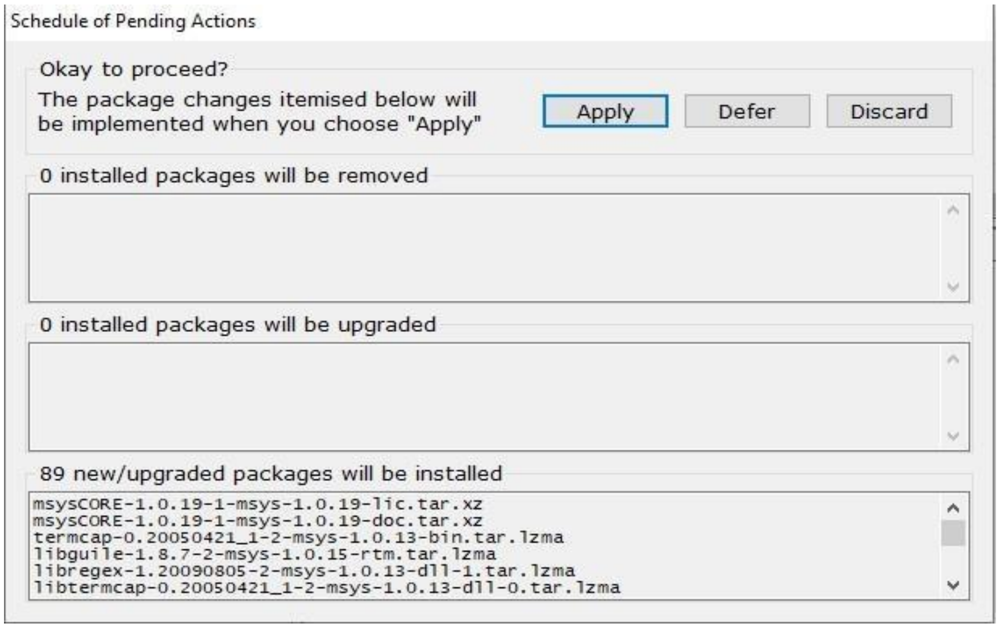


You have marked packages for installation, upgrade, or removal, but you have not yet committed these changes; if you quit without committing these changes, they will be discarded, and you may need to reschedule them.

You are advised to review your marked changes, before you quit; (you may select the "Review Changes" option button below), or alternatively, you may simply discard the changes, unseen, or you may cancel this request to quit.

7. Click **Review Changes**

   The following pop-up window should appear.



The package changes itemised below will be implemented when you choose "Apply"

0 installed packages will be removed

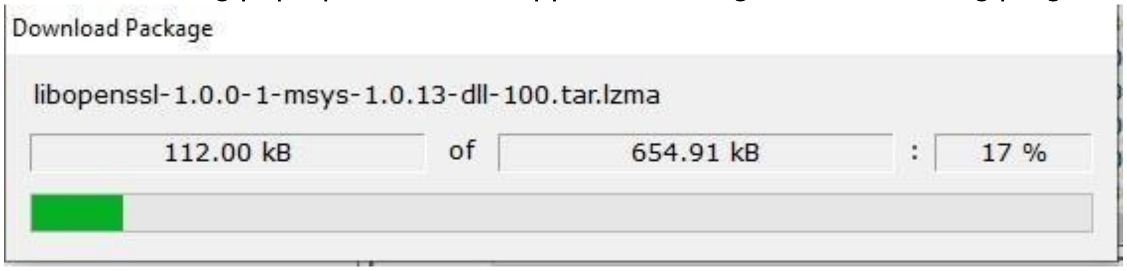
0 installed packages will be upgraded

89 new/upgraded packages will be installed

```
msysCORE-1.0.19-1-msys-1.0.19-lic.tar.xz
msysCORE-1.0.19-1-msys-1.0.19-doc.tar.xz
termcap-0.20050421_1-2-msys-1.0.13-bin.tar.lzma
libguile-1.8.7-2-msys-1.0.15-rtm.tar.lzma
libregex-1.20090805-2-msys-1.0.13-dll-1.tar.lzma
libtermcap-0.20050421_1-2-msys-1.0.13-dll-0.tar.lzma
```

8. Click **Apply**

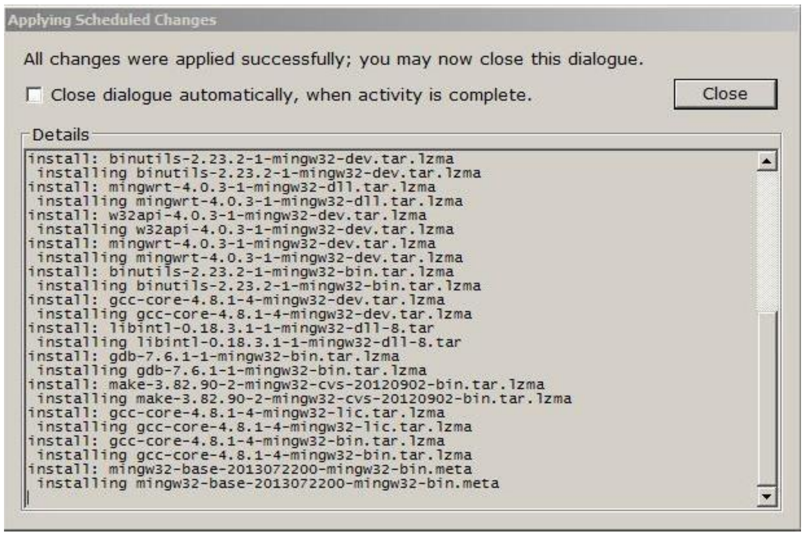
The following pop-up window will appear, showing the downloading progress.



libopenssl-1.0.0-1-msys-1.0.13-dll-100.tar.lzma

112.00 kB of 654.91 kB : 17 %

After a while (a few minutes to an hour, depending on your download speed), it shouldstart extracting the downloaded files. A few minutes after that, the following pop-up window should appear.



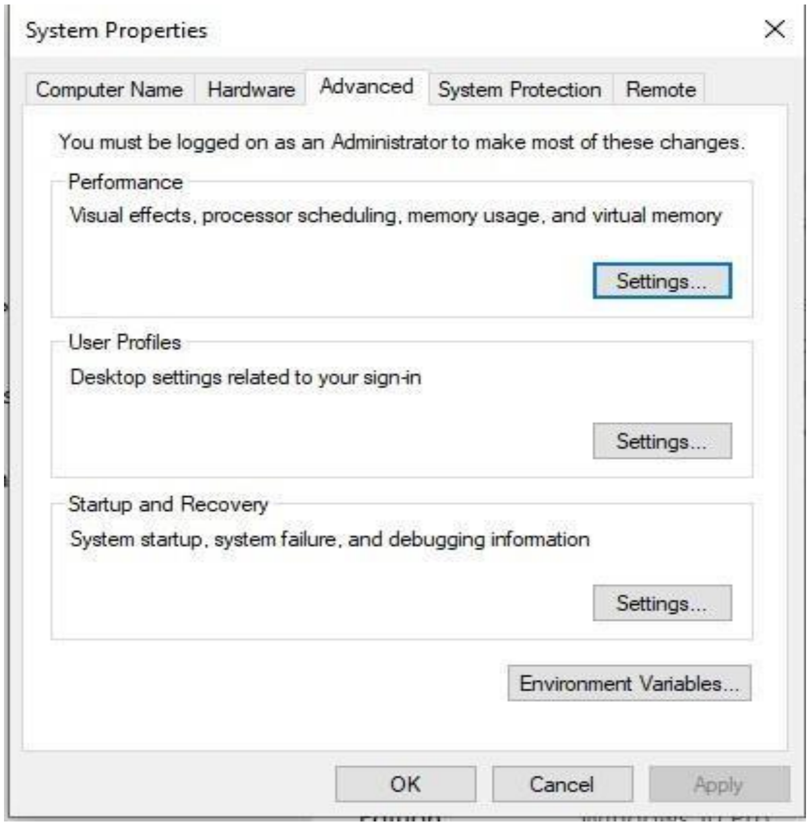
All changes were applied successfully; you may now close this dialogue.

☐ Close dialogue automatically, when activity is complete.

```
install: binutils-2.23.2-1-mingw32-dev.tar.lzma
 installing binutils-2.23.2-1-mingw32-dev.tar.lzma
install: mingwrt-4.0.3-1-mingw32-dll.tar.lzma
 installing mingwrt-4.0.3-1-mingw32-dll.tar.lzma
install: w32api-4.0.3-1-mingw32-dev.tar.lzma
 installing w32api-4.0.3-1-mingw32-dev.tar.lzma
install: mingwrt-4.0.3-1-mingw32-dev.tar.lzma
 installing mingwrt-4.0.3-1-mingw32-dev.tar.lzma
install: binutils-2.23.2-1-mingw32-bin.tar.lzma
 installing binutils-2.23.2-1-mingw32-bin.tar.lzma
install: gcc-core-4.8.1-4-mingw32-dev.tar.lzma
 installing gcc-core-4.8.1-4-mingw32-dev.tar.lzma
install: libintl-0.18.3.1-1-mingw32-dll-8.tar
 installing libintl-0.18.3.1-1-mingw32-dll-8.tar
install: gdb-7.6.1-1-mingw32-bin.tar.lzma
 installing gdb-7.6.1-1-mingw32-bin.tar.lzma
install: make-3.82.90-2-mingw32-cvs-20120902-bin.tar.lzma
 installing make-3.82.90-2-mingw32-cvs-20120902-bin.tar.lzma
install: gcc-core-4.8.1-4-mingw32-lic.tar.lzma
 installing gcc-core-4.8.1-4-mingw32-lic.tar.lzma
install: gcc-core-4.8.1-4-mingw32-bin.tar.lzma
 installing gcc-core-4.8.1-4-mingw32-bin.tar.lzma
install: mingw32-base-2013072200-mingw32-bin.meta
 installing mingw32-base-2013072200-mingw32-bin.meta
```

Click **Close.**

9. Edit Path so that the MinGW and MSYSM software is findable by VS Code.
    a. Click **This PC**
    b. Click **Local disk(C)**
    c. Click **MinGW**
    d. Click **bin**
    e. Copy **Path**
10. For setting path, Click **Properties** of **This PC**, then go to **Advance system setting**.



System Properties

Computer Name | Hardware | Advanced | System Protection | Remote

You must be logged on as an Administrator to make most of these changes.

Performance

Visual effects, processor scheduling, memory usage, and virtual memory

Settings...

User Profiles

Desktop settings related to your sign-in

Settings...

Startup and Recovery

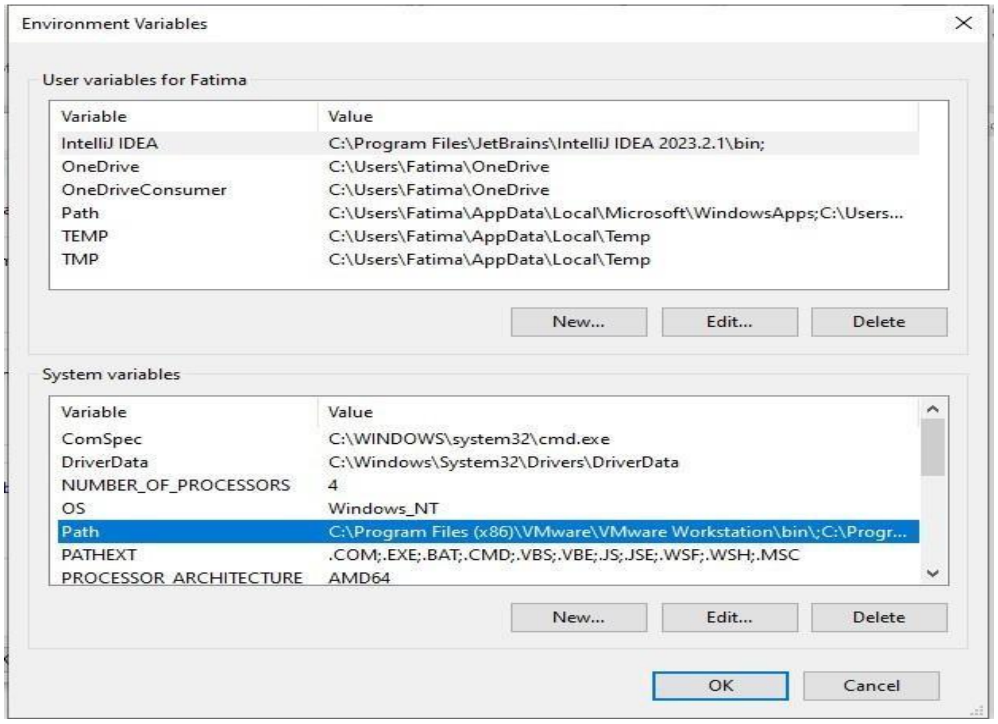
System startup, system failure, and debugging information

Settings...

Environment Variables...

OK Cancel Apply

11. Click Environment Variables. Following screen will appear on your screen.


Environment Variables

User variables for Fatima

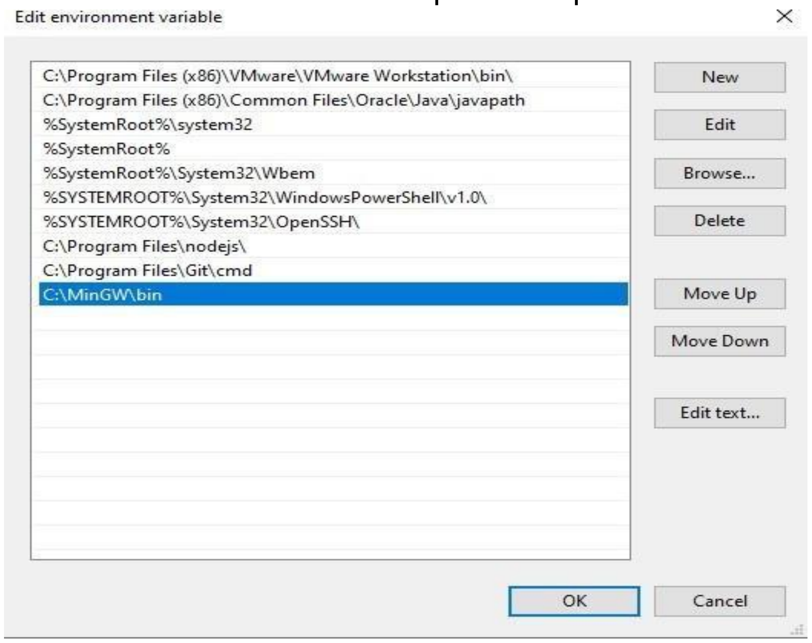
Variable	Value
IntelliJ IDEA	C:\Program Files\JetBrains\IntelliJ IDEA 2023.2.1\bin;
OneDrive	C:\Users\Fatima\OneDrive
OneDriveConsumer	C:\Users\Fatima\OneDrive
Path	C:\Users\Fatima\AppData\Local\Microsoft\WindowsApps;C:\Users...
TEMP	C:\Users\Fatima\AppData\Local\Temp
TMP	C:\Users\Fatima\AppData\Local\Temp

System variables

Variable	Value
ComSpec	C:\WINDOWS\system32\cmd.exe
DriverData	C:\Windows\System32\Drivers\DriverData
NUMBER_OF_PROCESSORS	4
OS	Windows_NT
Path	C:\Program Files (x86)\VMware\VMware Workstation\bin\;C:\Progr...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
PROCESSOR_ARCHITECTURE	AMD64

12. Select Path and then Click **Edit**.

13. Click **New** and then paste the path.

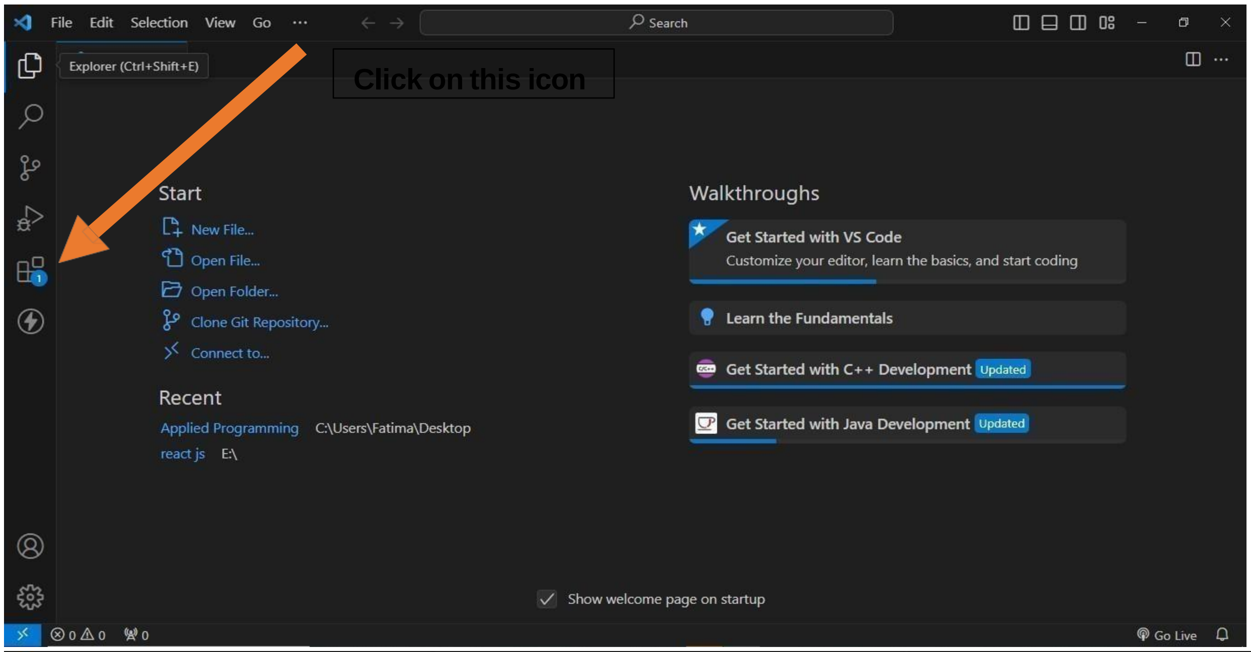

Edit environment variable

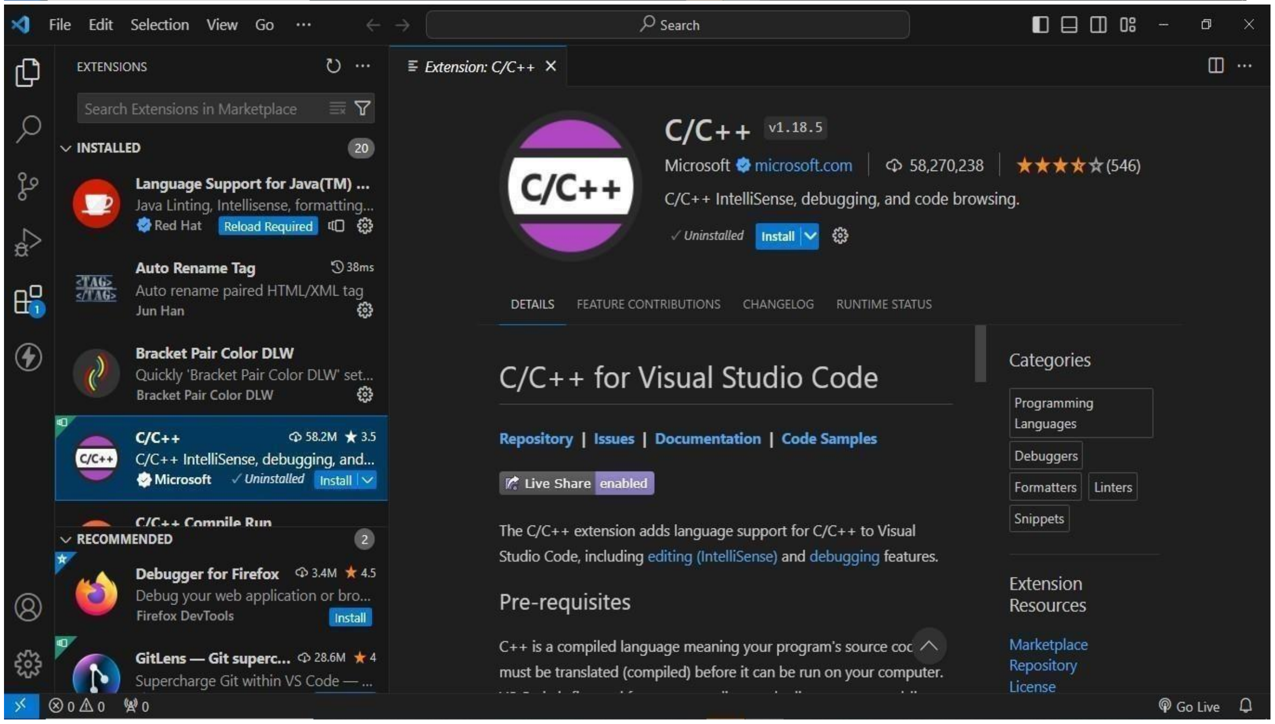
```
C:\Program Files (x86)\VMware\VMware Workstation\bin\
C:\Program Files (x86)\Common Files\Oracle\Java\javapath
%SystemRoot%\system32
%SystemRoot%
%SystemRoot%\System32\Wbem
%SYSTEMROOT%\System32\WindowsPowerShell\v1.0\
%SYSTEMROOT%\System32\OpenSSH\
C:\Program Files\nodejs\
C:\Program Files\Git\cmd
C:\MinGW\bin
```

14. Click **OK**

15. **MinGW** is now installed.

How to Install Extension in VS Code


Click on this icon
Explorer (Ctrl+Shift+E)
Start
New File...
Open File...
Open Folder...
Clone Git Repository...
Connect to...
Recent
Applied Programming C:\Users\Fatima\Desktop
react js E:\
Walkthroughs
Get Started with VS Code
Customize your editor, learn the basics, and start coding
Learn the Fundamentals
Get Started with C++ Development Updated
Get Started with Java Development Updated
Show welcome page on startup


EXTENSIONS
Search Extensions in Marketplace
INSTALLED 20
Language Support for Java(TM) ...
Auto Rename Tag
Bracket Pair Color DLW
C/C++ 58.2M 3.5
C/C++ IntelliSense, debugging, and...
Microsoft Uninstalled Install
RECOMMENDED 2
Debugger for Firefox
GitLens — Git superc...
Extension: C/C++
C/C++ v1.18.5
Microsoft microsoft.com 58,270,238 (546)
C/C++ IntelliSense, debugging, and code browsing.
Uninstalled Install
DETAILS FEATURE CONTRIBUTIONS CHANGELOG RUNTIME STATUS
C/C++ for Visual Studio Code
Repository | Issues | Documentation | Code Samples
Categories
Programming Languages
Debuggers
Formatters
Linters
Snippets

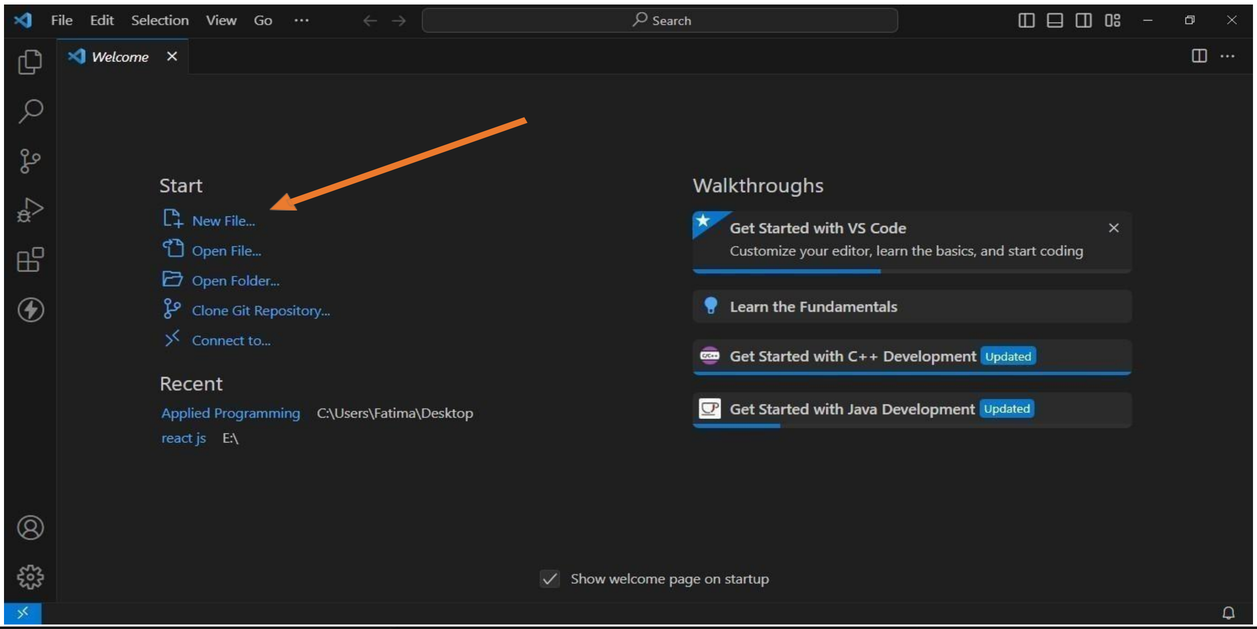
The C/C++ extension adds language support for C/C++ to Visual Studio Code, including editing (IntelliSense) and debugging features.

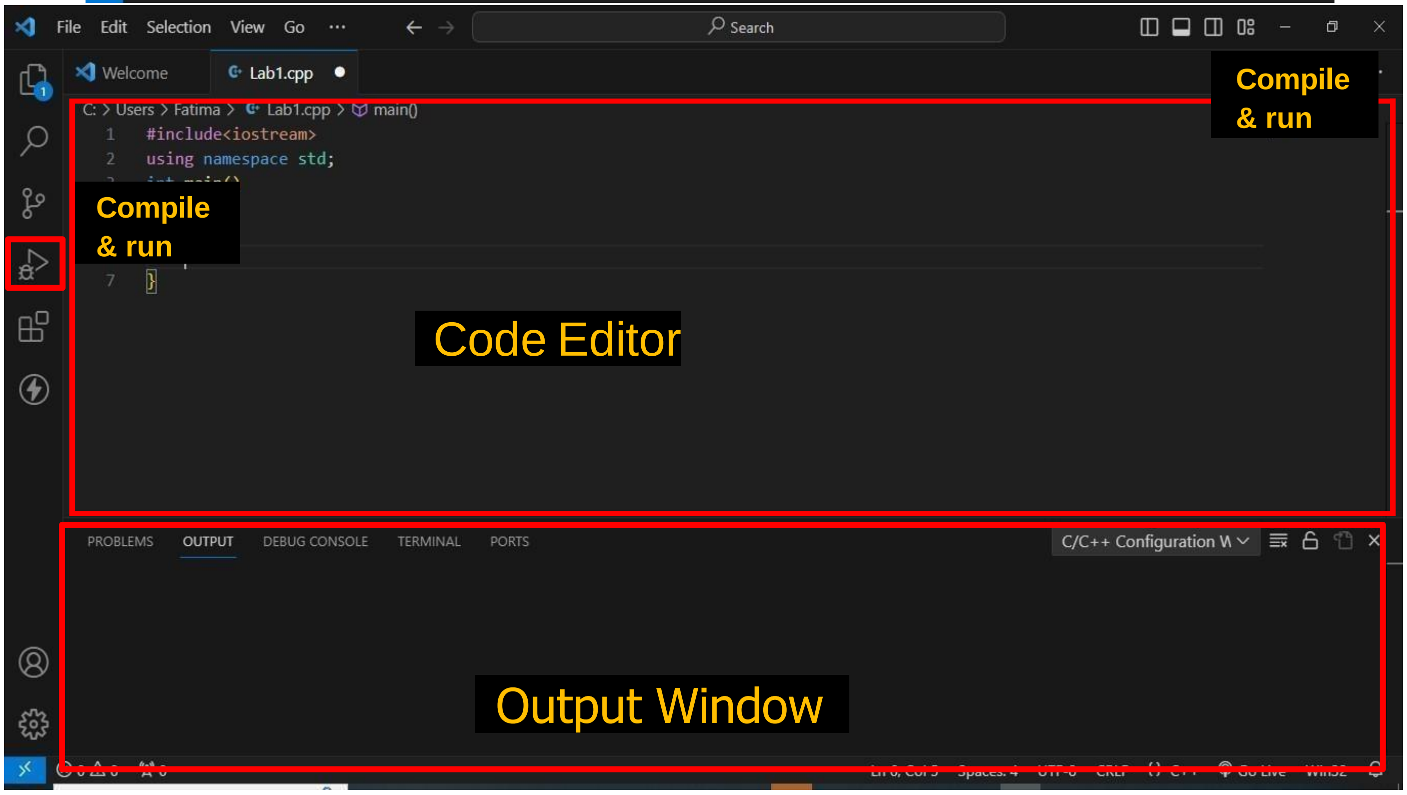
Pre-requisites

C++ is a compiled language meaning your program's source code must be translated (compiled) before it can be run on your computer.

CREATE NEW PROJECT

Create new and save this file with extension cpp i.e. file_name.cpp.


File Edit Selection View Go ···
Search
Welcome
Start
New File...
Open File...
Open Folder...
Clone Git Repository...
Connect to...
Recent
Applied Programming C:\Users\Fatima\Desktop
react js E:\
Walkthroughs
Get Started with VS Code
Customize your editor, learn the basics, and start coding
Learn the Fundamentals
Get Started with C++ Development Updated
Get Started with Java Development Updated
Show welcome page on startup



```
#include<iostream>
using namespace std;
```

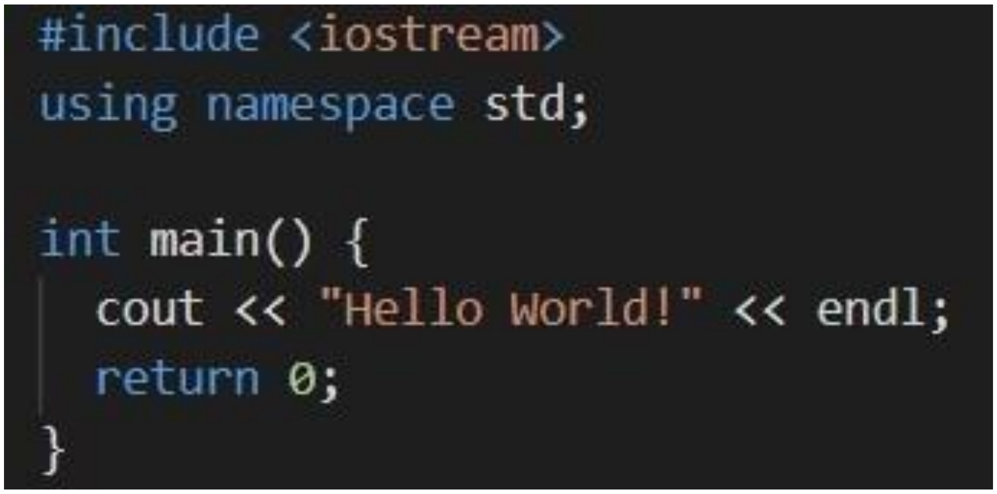
- **<u>Code Editor</u>**: Where the user will write code.
- **<u>Output Window</u>**: Here the Visual Studio shows the outputs, compiler warnings, error messages and debugging information.

SKELETON OF C++ PROGRAM

A C++ program is structured in a specific and particular manner. In C++, a program isdivided into the following three sections:

1. Standard Libraries Section
2. Main Function Section
3. Function Body Section

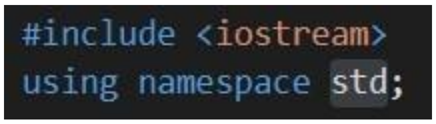
For example, let's look at the implementation of the Hello World program:



```
#include <iostream>
using namespace std;

int main() {
  cout << "Hello World!" << endl;
  return 0;
}
```

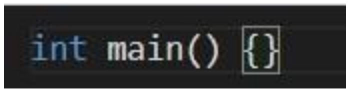
STANDARD LIBRARIES SECTION



```
#include <iostream>
using namespace std;
```

- #include is a specific preprocessor command that effectively copies and pastes the entire text of the file, specified between the angle brackets, into the source code.
- The file <iostream>, which is a standard file that should come with the C++ compiler, is short for input-output streams. This command contains code for displaying and getting an input from the user.
- namespace is a prefix that is applied to all the names in a certain set. iostream file defines two names used in this program - cout and endl.

MAIN FUNCTION SECTION



```
int main() {}
```

- The starting point of all C++ programs is the main function.
- This function is called by the operating system when your program is executed by the computer.
- {Signifies the start of a block of code, and} signifies the end.

INPUT/OUTPUT IN C++

- C++ is very similar to the C Language.
- For the input/output stream we use <iostream> library (in C it was <stdio>).
- For taking input and out we cout and cin (in C it was printf and scanf).
- cout uses insertion (<<) operator.
- cin uses extraction (>>) operator.

SAMPLE C++ CODE:

```
#include <iostream>
using namespace std;
int main()
{
int var = 0;
cout << "Enter an Integer value: ";
cin >> var;
cout << "Value of var is : " << var;
return 0;
}
```

Sample Run: In this sample run, the user input is shaded. Enter an Integer value: 12
Value of var is : 12

ARRAYS

An Array is a collection of fixed number of elements of same data type.

1-D ARRAY

- 1-D Array is a form of array in which elements are arranged in a form of List.
- To declare a 1D array you need to specify the data type, name and array size.

dataType arrayName [arraySize] ;

- Following is the declaration of a 1D array.

int numArray[5];

where;

Data Type: Integers (int)

Array Name: numArray

Array Size: 5

- To access array elements, you use the array name along with the index in subscript operator **"[ ]"**.

numArray[0], numArray[1], numArray[2], numArray[3], numArray[4]

where;

Index of the array starts with zero '0'.

Index of the last element is always 'size - 1' (in this case it is 4).

Example Code for 1-D Array

Program to read five numbers, find their sum, and print the numbers in reverse order.

```
#include <iostream>
using namespace std;
int main()
{
int item[5]; //Declare an array item of five components int sum = 0;
int counter;
cout << "Enter five numbers: ";
for (counter = 0; counter < 5; counter++)
{
cin >> item[counter];
}
for (counter = 0; counter < 5; counter++)
{
sum = sum + item[counter];
}
cout << endl;
cout << "The sum of the numbers is: " << sum << endl; cout << "The numbers in reverse order
are: ";
//Print the numbers in reverse order.
for (counter = 4; counter >= 0; counter--)
cout << item[counter] << " ";
cout<<endl;
return 0;
}
```

2-D ARRAY

1. 2-D Array is a collection of fixed collection of elements arranged in rows and columns.
2. To declare a 2D array you need to specify the data type, name and no. of rows and columns. **dataType arrayName [rowSize][columnSize] ;**
3. Following is the declaration of a 2D array.

   int numArray[5][5];

4. To access array element you use the array name along with the row Index and column Index in subscript operator **"[ ][ ]".**

   numArray[0][0], numArray[1][1], numArray[2][2], numArray[3][3], numArray[4][4].

   where; Index for the rows and columns of the array starts with zero '0'.

   Index of the last element in rows and columns is always **'sizeofRow - 1'** and **'sizeofColumn -1'** respectively (in this case it is 4).

Example Code for 2-D Array:

Program to read a 2D array of size 3x3 find the sum for each row, print the sum line by line.

```
#include <iostream>
using namespace std;
int main()
{
int item[3][3];      //Declare an array of size 3x3

int row, col;
cout << "Enter array elements: " << endl;
for (row = 0; row < 3; row++)
{
   int sum = 0;
for (col = 0; col < 3; col++)
{
cin >> item[row][col];
sum = sum + item[row][col];
}
cout << "The sum of row " << row << " : " << sum <<
endl;
}
cout << endl; return 0;
}
```

Sample Run: In this sample run, the user input is shaded.

```
Enter array elements:
12 76 34
The sum of row 0: 122
52 89 48
```

```
The sum of row 1: 189

22 63 99

The sum of row 2: 184
```

DYNAMIC MEMORY ALLOCATION

1. Variables created during the program execution are called dynamic variables.To create a dynamic variable, we use new operator.

   new dataType [size]; // to allocate an array of variables.

2. The new operator allocates the memory of a designated type.It returns a pointer to the allocated memory.
   Following is the declaration of a dynamic variable.
   int *p = new int;
   char *cArray = new char[5];

.

3. To delete the dynamically allocated memory we use delete operator. **delete ptrVar; //to deallocate single dynamic variable**
   delete [] ptrArray; //to deallocate dynamically created array.

4. delete operator is used to free the memory which is dynamically allocated using newoperator.

Example Code for Dynamic Variables

```
#include<iostream>
using namespace std; int main()
{
int* intPtr;
char* charArray;
int arraySize;
intPtr = new int; // Allocating memory for a single variable
cin >> arraySize;
charArray = new char[arraySize]; // Allocating memory for the array
// Input values into the character array
 for (int i = 0; i < arraySize; i++)
cin >> charArray[i];
// Output the values in the character array
for (int i = 0; i < arraySize; i++)
cout << charArray[i];
delete[] charArray;
 return 0;
}
```

Sample Run: In this sample run, the user input is shaded.

```
Enter an Integer Value: 2
Enter the size of the Character Array : 2 a b
ab
```

POINTERS

A Pointer is a variable whose content is a memory address.

SINGLE POINTERS

1. To declare a single pointer variable you need to specify the data type, an asterisk symbol (*) and the name of the pointer variable.

   dataType *ptrName;

2. Following is the declaration of a Pointer variable.

   int *ptr;

3. Pointer variable holds the memory address of the variable which is of same data type (integer in this case).
4. To assign the memory address of any variable to the pointer variable we use Address of Operator (&).

   int intVar = 5;
   ptr= &intVar;

   In this statement ptr now holds the memory address of an integer variable 'intVar'.

5. To access the value at the memory address (currently stored) in the variable we use Dereferencing Operator (*).

6. Do not confuse this with the symbol used for the declaration of a pointer.

int intVar2 = *ptr;

In this statement another integer variable ‘intVar2’ is now initialized with the value at the memory address which is stored in ptr (that is the value of intVar).

Example Code for Single Pointer with DMA

The following program illustrates how pointer variables work:

```
#include <iostream>
using namespace std;

int main() {
    // Allocate memory for an integer using DMA
    int *p = new int;

    // Assign a value to the dynamically allocated integer
    *p = 37;

    cout << "Line 1: *p = " << *p << endl;

    // Modify the value through the pointer
    *p = 58;

    cout << "Line 3: *p = " << *p << endl;

    // Deallocate the dynamically allocated memory
    delete p;

    return 0;
}
```

Sample Run:

```
Line 1: *p = 37

Line 3: *p = 58
```

2D Pointers:

In C++, a 2D array can be represented using pointers. A 2D pointer is essentially a pointer to an array of pointers, where each pointer points to a separate array representing a row in the 2D array.

Declaration:

```
int** matrix; // Declare a 2D pointer
```

Dynamic Allocation:

```
int rows = 3;
int cols = 4;

matrix = new int*[rows]; // Allocate memory for rows

for (int i = 0; i < rows; ++i) {
    matrix[i] = new int[cols]; // Allocate memory for each row
}
```

Example Code for 2D Pointers:

```
#include <iostream>
using namespace std;
int main() {
    int rows = 3;
    int cols = 4;
    int** matrix = new int*[rows];
   for (int i = 0; i < rows; ++i) {
        matrix[i] = new int[cols];
    }
    // Access and modify elements
    matrix[0][0] = 10;
    matrix[1][2] = 20;
    // Display elements
    for (int i = 0; i < rows; ++i) {
        for (int j = 0; j < cols; ++j) {
            cout << matrix[i][j] << " ";
        }
        cout << endl;
    }
    // Deallocate memory
    for (int i = 0; i < rows; ++i) {
        delete[] matrix[i];
    }

    delete[] matrix; return 0;}
```

Lab Tasks:

Q1 Write a program in c++ for fund deposit and fund withdraw. The user can both deposit and withdraw amount from account. If the user enters incorrect amount, a message will be displayed indicating insufficient funds at that time.

For Example:

```
Welcome to the Banking System
1. Deposit Funds
2. Withdraw Funds
3. Exit
Enter your choice: 1
Enter the amount to deposit: 1000
Deposit successful. Your new balance is: 1000
Welcome to the Banking System
1. Deposit Funds
2. Withdraw Funds
3. Exit
Enter your choice: 2
Enter the amount to withdraw: 500
Withdrawal successful. Your new balance is: 500
```

Q2 Create a BMI calculator application that reads the user's weight in pounds and height in inches

then calculates and displays the user's body mass index.

Formula for calculating BMI:

```
BMI= (weight_pounds / (pow(height_inches, 2))) * 703;
```

Q3 Write a Function to find the largest number and the second largest number from an array. However a function cannot return multiple value. We need to use pointers to implement the function as follow:

```
void find_two_largest(int a[], int *largest, int *second_largest);
```

When passing an array a of length n, the function will search for its largest and second largest elements, storing them in the variable pointed to largest and second_largest.

Q4 Write a C++ program to declare an array of size 10, prompt the user to input the elements of the array, and replace all even numbers with 1 and odd numbers with 'o'.

Q5 Write a program in C++ to perform matrix subtraction for order 3x3 matrices using 2D array.