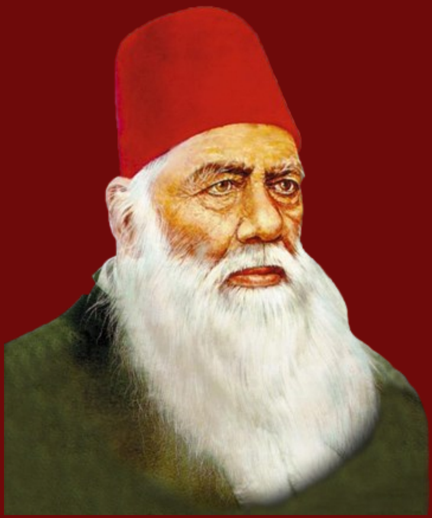
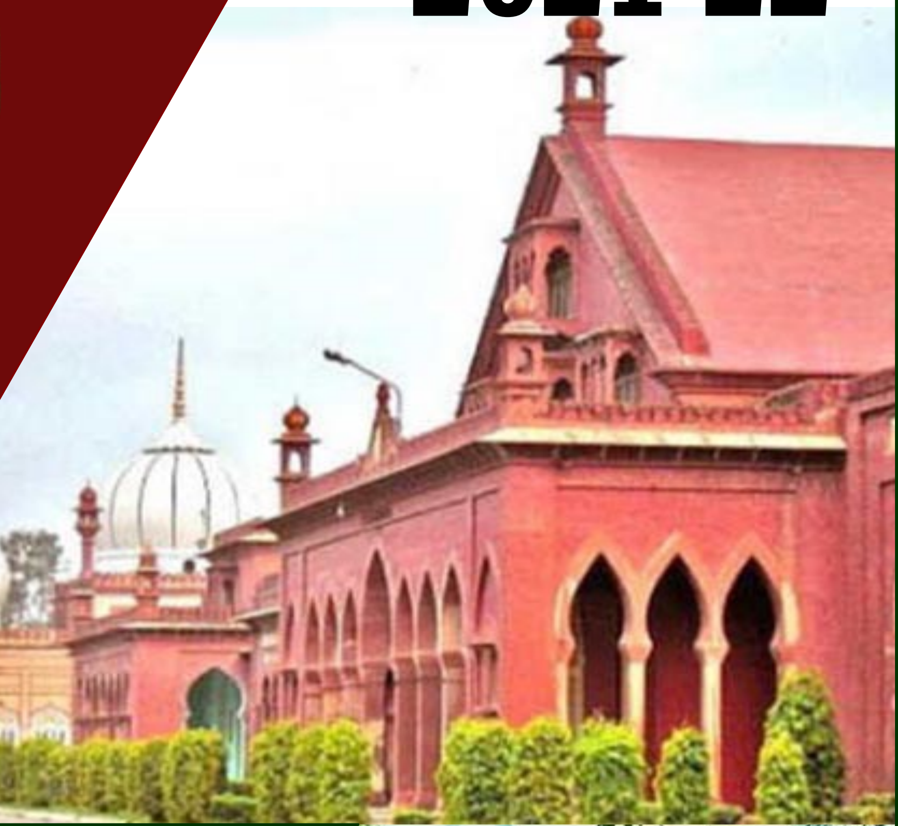


*Look forward,
learn modern
knowledge, and
do not waste
time in studies
of old subjects of
no values.*

2021-22



Computer Lab Manual

M.C.A.
Semester-I

CSD-10P1: LABORATORY COURSE – I

DEPARTMENT OF COMPUTER SCIENCE

ALIGARH MUSLIM UNIVERSITY ALIGARH

Important Instructions to the students in the ongoing Covid-19 Scenario

❒ **In the present Covid-19 scenario, all the labs will be conducted ONLINE till the further instructions issued by the University.**

❒ **Students are instructed to perform their lab exercises/assignments at their own system from their respective places.**

❒ **During this prevailing situation (COVID-19 crises), the mode (policies/rules) of conducting Lecture/Lab classes will be as per the instructions issued by the University from time to time.**

❒ **The students are advised to complete the weekly activities/assignments well in time (i.e., within the same week).**

❒ **The students are also advised to maintain the soft copy of the Lab File of their completed activities/assignments in the prescribed format.**

!! Stay Home, Stay Safe !!

Lab Manual: Laboratory Course – I (CSD10P1)

CONTENTS **PAGE NO.**

WEEK WISE CONTENTS

COURSE TITLE: LABORATORY COURSE-I	**COURSE CODE:** CSD – 10P1
CREDIT: 4	**PERIODS PER WEEK:** 6
CONTINUOUS ASSESSMENT: 40 Marks	**EXAMS:** 60 Marks

COURSE DESCRIPTION

Knowledge in a programming language is prerequisite to the study of most of computer science courses. This knowledge area consists of those skills and concepts that are essential to problem solving and programming practice independent of the underlying paradigm. All programming languages will share one similarity, i.e., all are based on logic. A programming language is a programmer’s principal interface with the computer. More than just knowing how to program in a single language, programmers need to understand the different styles of programming promoted by different languages. Later in your professional life, you will be working with many different programming languages and styles at once, and will encounter many different languages over the course of your career. Understanding the variety of programming languages and the design trade-offs between the different programming paradigms make it much easier to master new languages quickly.

This course examines practical programming techniques and issues, emphasizing object modelling and simulation. The objectives of the course are to explore issues involved in developing large-scale object-oriented systems and to teach fundamental techniques that can simplify software development. The course provides in depth information on object oriented programming, issues, techniques, and methodologies.

CONTENT

This course introduces to fundamental 'computer literacy' concepts. The objective is to explore the knowledge of fundamental of information technology and information system.

This course is designed to provide the students the opportunity of learning both – concepts of C++ and then learning the object-oriented concepts using C++. Our goal will be to improve your reasoning and thinking skills, which should prove helpful, not only in future programming, but throughout your academic and professional career.

OBJECTIVES

This course is designed to help students in:

- ❒ To help the students in learning the C++.
- ❒ To help the students how to write a sample program in C++.
- ❒ To help the students in understanding the use of different variables.
- ❒ To help the students in understanding the use of decision-making statement.
- ❒ Be able to debug and test C++ programs.
- ❒ Understand how to read C++ library documentation & reuse library code.
- ❒ To make the students understand the features of object-oriented principles and familiarize them with virtual functions, templates and exception handling.

OUTCOMES

After completing this course, the students would be able to:

- ❒ Write, debug and run a sample program in C++.
- ❒ Differentiate the variables and constants.
- ❒ Understand how decision making statements are written.

❒ Create a simple real life applications based on the decision making statements.

❒ Understand the concepts and implementation of constructors and destructors.

❒ Develop software applications using object-oriented programming language in C++.

❒ Understand and use the basic programming constructs of C++.

❒ Learn C++ data types, memory allocation/reallocations, functions and pointers.

❒ Apply object-oriented programming concepts to software problems in C++.

HOW TO DO WELL IN THIS COURSE

❒ The students are advised to attend all their theory classes and respective labs regularly as both are integrated to each other. If any student will miss the theory lecture, he/she may not be able to do well in lab related to that topic.

❒ The students are advised to submit the assignments given in theory and lab classes timely to their respective Teachers/Instructors online.

❒ The students should demonstrate disciplined and well-behaved demeanor in the Department.

❒ Each student shall be assigned a system in their introductory lab. They are advised to do their work on that system only for the whole semester. Students should store all their lab activities regularly.

❒ All students are advised to understand course objectives and outcomes and achieve both during their lab work.

❒ The students are advised to follow books/eBooks/online tutorial/other online study material links given in lecture/lab manual/ syllabus references. These study materials are very helpful in terms of skills, knowledge and placement.

- ❒ This Lab course is very important in terms of placement. Therefore, students are advised to implement all the problems by her /him given in the individual week.
- ❒ All students are advised to solve old placement papers for campus selection. Following links may be useful for the preparation of your campus placements.
  - ➢ https://www.indiabix.com/placement-papers/companies/
  - ➢ https://www.offcampusjobs4u.com/download-tcs-placement-test-question-papers-with-solutions/
  - ➢ https://www.indiabix.com/placement-papers/tcs/
  - ➢ https://www.firstnaukri.com/career-guidance/infosys-placement-papers-with-solutions-2019-firstnaukri-prep
  - ➢ https://prepinsta.com/ibm/
  - ➢ https://www.faceprep.in/infosys/infosys-aptitude-questions/
  - ➢ https://alpingi.com/infosys-placement-papers-solution-pdf-download/
  - ➢ http://placement.freshersworld.com/
- ❒ The students are advised to follow mentioned tutorials links:
  - ➢ https://www.javatpoint.com/cpp-tutorial
  - ➢ https://www.tutorialspoint.com/cplusplus/index.htm
  - ➢ http://www.cplusplus.com/doc/tutorial/
- ❒ The students are advised to follow below Links for installing application software:
  - ➢ http://www.bloodshed.net/dev/devcpp.html
  - ➢ http://www.codeblocks.org/downloads
- ❒ The students are advised to use the following online editors which are much helpful during the online classes:
  - ➢ **JDOODLE Online C++ Compiler**

https://www.jdoodle.com/online-compiler-c++/

- **Online GDB Online C++ Compiler**

  https://www.onlinegdb.com/online_c++_compiler

- **C++ Shell Online C++ Compiler**

  http://cpp.sh/

- **Programiz C++ Online Compiler**

  https://www.programiz.com/cpp-programming/online-compiler/

RULES AND REGULATIONS

Students are required to strictly adhere to the following rules.

- ❒ The students must complete the weekly activities/assignments well in time (i.e., within the same week).
- ❒ The students must maintain the Lab File of their completed activities/assignments in the prescribed format (**Appendix-1**).
- ❒ The students must get the completed weekly activities/assignments checked and signed by the concerned teachers in the Lab in the immediate succeeding week. Failing which the activities/assignments for that week will be treated as incomplete.
- ❒ At least **TEN (10)** such timely completed and duly signed weekly activities/assignments are compulsory, failing which students will not be allowed to appear in the final Lab Examination.
- ❒ The students need to submit the following three deliverables for each exercise duly signed by the Teacher:
  - ❖ Coding
  - ❖ Input /Output
- ❒ The students need to ensure that each question is assessed and signed by the Teacher in the week/time.

❒ Late submission would not be accepted after the due date.

❒ Cooperate, collaborate and explore for the best individual learning outcomes but copying is strictly prohibited.

APPENDIX–I

Template for the Index of Lab File

WEEK NO.	PROBLEMS WITH DESCRIPTION		PAGE NO.	SIGNATURE OF THE TEACHER WITH DATE
1	1#			
	2#			
	3#			
2	1#			
	2#			
	3#			
3	1#			
	2#			
	3#			

Note: The students should use Header and Footer mentioning their roll no. & name in footer and page no. in header.

WEEK #1

OBJECTIVES

- ❒ To help the students in learning *the practical use of MS-Word, MS-Excel.*

OUTCOMES

After completing this, the students would be able to:

- ❒ Use MS-Word, MS-Excel in real life applications.

PROBLEMS

1# Open a new document and type the following letter.

July 15, 2020,
Chennai

From
VENKATESH .P
Sri Ranga Apartments,
No: 120, II Avenue,
T. Nagar. Chennai-17
To
<<Name>>
<<Address>>

Respected <<Name>>

With the current slowdown in hiring within the high- tech field, you must be flooded with resumes from out-placed software engineers such as me. Please take a moment to consider my qualifications. I believe in particular is highly marketable in this tight market:

I worked on the team that pioneered the technology that put the Palm Pilot on the map.

In today's increasingly mobile society, this technology has places to go, and I have ideas that could take us to the next step in office independence.

Please call me with prospective job opportunities. I am interested in a project management position in the Rs. 9K range.

Thank you!

Venkatesh.

Enclosure: Resume (the format given below).

i) Save the document as "Letter.doc."

ii) Send the document to 3 recipients using Mail merge. (Use 3 different addresses)

iii) Define a Macro 'Decorate' which makes the text bold, Red in color and italic, font size Assign a shortcut key Alt + Z to this macro.

iv) Close the document.

v) The Sample Addresses are:

i) Mr. Amit Tandon
13, New Estate,
Ring Road, Chandigarh

ii) Mr. Rohit Saluja
15, Karol Bagh, New Delhi

iii) Ms. Jyoti Parmar
Sector 16, New Building, Gurugram

Format of Resume

Name :

Father's Name :

Date of Birth :

Age :

Address :

Educational Qualification :

Sr.	Qualification	Board/University	Percentage

Work Experience :

Technical Skills :

Personal Skills :

Hobbies :

Dated:…………………. Signature

2# Create a table in word as shown below:

Roll No	Name	Marks in Physics	Marks in Chemistry	Total Marks
1	Sakshi	80	70	
2	Rohit	70	80	
3	Amit	60	50	
4	Rakesh	40	60	
5	Komal	30	70	
6	Garima	80	80	

Do the following:

(a) In the total marks column, entries should be calculated using formulas and it is the sum of marks in physics and marks in chemistry.

(b) Insert a new row at the end of the table and also find grand total using formula.

(c) Sort the table based on total marks.

(d) The date and heading should be centre aligned.

(e) Heading should be in bold and underlined.

3# Using a spreadsheet package you have studied, construct T Morongo's pay slip for December 2016 following the instructions below. Insert a custom footer with your *name, subject, course, exam/Test &question number*. Save it as Salary advice.

SALARY ADVICE FOR MARCH 2016			
EMPLOYEE	T MARONGO		
STAFF NO	004		
DATE	31 MARCH 2016		
NEXT PAY DATE	30 APRIL 2016		
BASIC SALARY p.a.	31200.00		
INCOME	AMOUNT	DEDUCTIONS	AMOUNT
Basic Salary		Pension @8%	
Housing Subsidy		P.A.Y.E.	
Vehicle Allowance		U.I.F.	
		Medical Aid	
		Bond Repayment	
Gross Income		Total Deductions	
Net Salary			

Instructions:

i) Housing Subsidy 6000.00 per year.

ii) Car Allowance 100.00 per month,

iii) Pension 8% on Basic Salary.

iv) P.A.Y.E. 636.83

v) Medical Aid 70.00

vi) U.I.F. 1% on Basic Salary + Housing Subsidy

vii) Bond Repayment 630.00

viii) Calculate Net Salary.

ix) Format all figures to two decimal places and insert ₹ currency symbol.

x) Insert a custom footer with your name, subject, and question number. Save it as salary advice2.

4# Use a new workbook & construct a worksheet with the data given & save it as LYONS.

LYONS INC

Orange JUICE Sales

PRODUCT	**COST PRICE PER LITRE**	**MARK UP PER ITEM 35%**	**SELLING PRICE**	**LITRES SOLD**	**TOTAL INCOME**	**PROFIT**
Cascade	3.75			234		
Quench	3.65			345		
Xtra	4.25			456		
Sun Splash	1.50			123		
House Brand	1.50			245		
TOTAL						
HIGHEST						
LOWEST						

Instructions

- ✓ **Markup** = Cost price/Litre x 35%
- ✓ **Selling price**= Cost price/Litre + Mark up
- ✓ **Total income**= Litres sold x Selling Price
- ✓ **Profit** = Total income – (Cost price/Litre x Litres sold)

a) The MARKUP % (35%) must be inserted in a separate cell under the heading. USE IT as an absolute cell reference in the formula to calculate the mark up per item.

b) Calculate the mark up for each item.

c) Calculate the selling price for each item.

d) Calculate the Total Income for each item.

e) Calculate the profit for each item.

f) Format the column LITRES SOLD to display the number of litres as integers. The rest of the worksheet must be formatted to display two decimals.

g) Use statistical functions to calculate the:

- ✓ AVERAGE
- ✓ HIGHEST (MAX)
- ✓ LOWEST (MIN) for Selling Price column up to Profit Column.

h) Show all formulas you have used in a new sheet. Adjust the column width so that the formulae are displayed in full and the sheets fits into one side of A4 landscape format and save it as formulas.

i) Under the worksheet Create a pie chart titled PRODUCT COST PER UNIT for Product & Cost price per Litre columns. Data labels indicating percentages should be displayed.

j) Put borders neatly on the on the work sheet & save it as LYONS2.

WEEK #2

OBJECTIVES

- ❒ To help the students in learning about MS-Power Point.

OUTCOMES

After completing this, the students would be able to:

- ❒ Understand the usage of MS-Power Point in real life.

PROBLEMS

1# Design Seasonal Greeting cards using MS-Power Point.

2# Design a AMU Magazine cover in MS-Power Point. Use the following:

i) Select a theme for the page,

ii) Insert either a picture or clipart, and

iii) Use WordArt.

3# Design a poster inviting all students of your department to the IT Fest (using MS-Power Point).

4# Create a 5-slide presentation on any topic. Use Images, Graphs, Chart, Tables, Animation, Time, Bullets, Transition, Sound, Hyperlink, Background template, Header and Footer (using MS-Power Point).

5# Create a 5-slide presentation on any topic. Use Images, Graphs, Chart, Tables, Animation, Time, Bullets, Transition, Sound, Hyperlink, Background template, Header and Footer (using MS-Power Point).

WEEK #3

OBJECTIVES

- To help the students in learning the concepts of C++.
- To help the students in learning the different tools for C++ programming on different software platform.
- To help the students in installing the C++ tool kit in their computers.

OUTCOMES

After completing this, the students would be able to:

- Understand the concepts of C++.
- Setup the C++ environment on different platform.

Dev C++ IDE: Installation, Features and C++ Development

Dev-C++ is a fully featured graphical IDE (Integrated Development Environment) that uses the MinGW compiler system to create Windows as well as Console based C/C++ applications. It can also be used with any other GCC-based compiler like Cygwin.

Dev-C++ is free software and is distributed under the GNU General Public License. Thus we can distribute or modify the IDE freely. It was originally developed by “Bloodshed Software”. It has been formed by Orwell after it was abandoned by Bloodshed in 2006.

Section 1.01 Features of Dev-C++ IDE

Enlisted below are some of the features of this IDE that help us developing efficient and user-friendly C/C++ applications.

- Dev-C++ supports GCC-based compilers including Cygwin, MinGW, etc. We can either install a Dev-C++ IDE along with the compiler integrated or just an IDE if we already have a compiler on our system.
- We can use integrated debugging (using GDB) with this IDE. The debugger allows us to perform all the general debugging operations on source code.
- It has a localization feature that provides support for multiple languages. We can select the language the first time when we open the IDE after installing it. We can also change the language anytime using settings.
- Like the other IDEs, this IDE also provides the "Auto-Completion" feature for the code we write.
- It comes with customizable syntax highlighting editor that can make the source code more readable.
- Allows editing and compiling the Resource files.
- Has a Tool Manager that contains various tools that can be used in the project.
- This IDE also has inbuilt Find and replace facilities.
- Using Dev-C++ IDE, we can create various types of applications be it Windows, Console, Static libraries or DLLs.
- We can also create our own project templates to create our own project types.
- Make files that are used for managing the build process for the application can also be created using dev-C++ IDE.
- It provides support for Class Browser as well as Debug variable Browser.
- It has a Project Manager that helps us to manage various projects.
- Also provides print support through its interface.
- We can easily install the add-on libraries using the package manager provided by the IDE.
- This C++ IDE also provides CVS support for source code management.

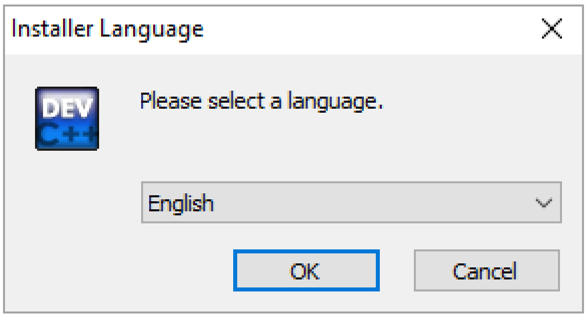
Section 1.02Installing and Configuring C++ IDE

We can get the appropriate installable for dev-C++ IDE from http://www.bloodshed.net/dev/devcpp.html

Let's see the entire installation process now. We have used the installable that comes along with the C++ compiler. In this tutorial, we use the dev-C++ version 5.11 with the TDM-GCC 4.9.2 compiler.

The stepwise installation for dev-C++ is given below.

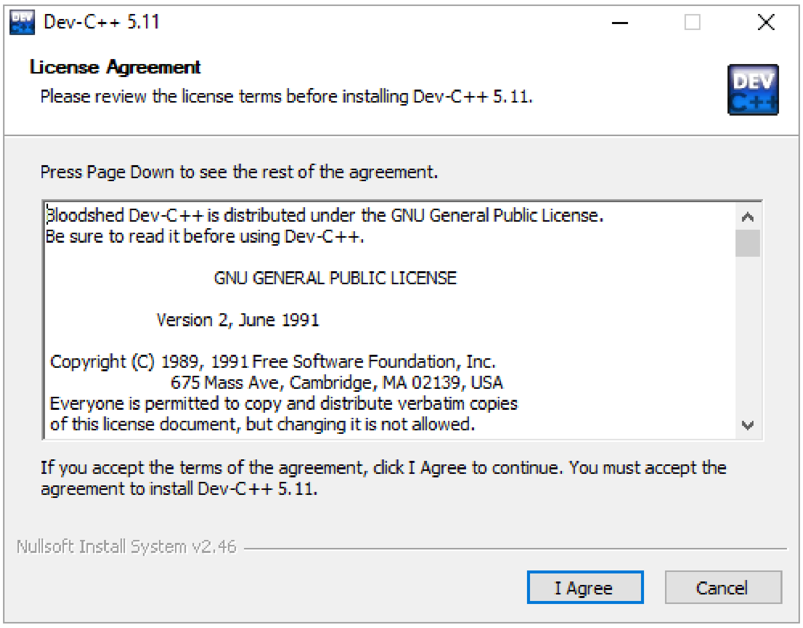
1) The first step while we start the installer is to select the language of our choice as shown in the below screenshot.



Please select a language.

English

2) Once you select the appropriate language, you have to agree to the license agreement that pop-ups next.



License Agreement

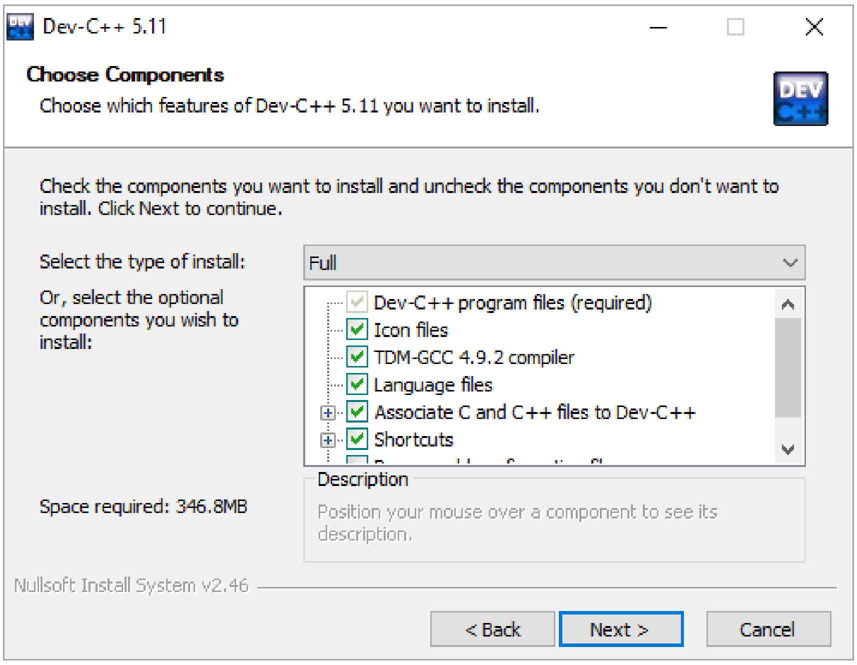
Please review the license terms before installing Dev-C++ 5.11.

Press Page Down to see the rest of the agreement.

```
Bloodshed Dev-C++ is distributed under the GNU General Public License.
Be sure to read it before using Dev-C++.

                        GNU GENERAL PUBLIC LICENSE

                Version 2, June 1991

 Copyright (C) 1989, 1991 Free Software Foundation, Inc.
                675 Mass Ave, Cambridge, MA 02139, USA
 Everyone is permitted to copy and distribute verbatim copies
 of this license document, but changing it is not allowed.
```

If you accept the terms of the agreement, click I Agree to continue. You must accept the agreement to install Dev-C++ 5.11.

3) Next, we are asked to select the components that we need to install as a part of the dev-C++ installation.



Dev-C++ 5.11

Choose Components

Choose which features of Dev-C++ 5.11 you want to install.

Check the components you want to install and uncheck the components you don't want to install. Click Next to continue.

Select the type of install: Full

Or, select the optional components you wish to install:

- ☑ Dev-C++ program files (required)
- ☑ Icon files
- ☑ TDM-GCC 4.9.2 compiler
- ☑ Language files
- ☑ Associate C and C++ files to Dev-C++
- ☑ Shortcuts

Description

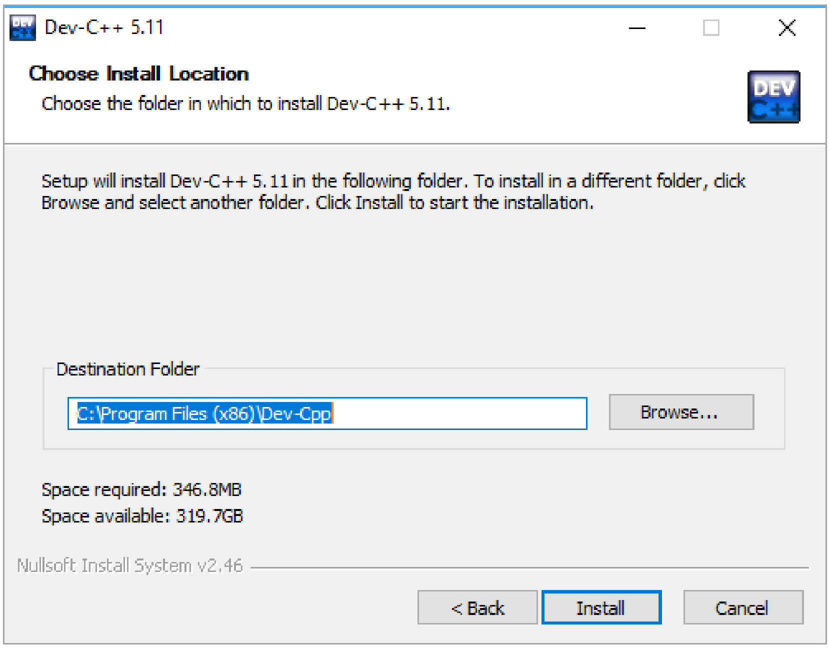
Position your mouse over a component to see its description.

Space required: 346.8MB

Nullsoft Install System v2.46

As shown in the above screenshot, we are provided with a list of components available for installation and a checkbox against each component. We can check/uncheck each box to indicate which components to install. Click next once the components are selected.

4) Now the installer prompts the user for the destination folder where the dev-C++ files/libraries etc. are to be copied.



Dev-C++ 5.11

Choose Install Location

Choose the folder in which to install Dev-C++ 5.11.

Setup will install Dev-C++ 5.11 in the following folder. To install in a different folder, click Browse and select another folder. Click Install to start the installation.

Destination Folder

C:\Program Files (x86)\Dev-Cpp

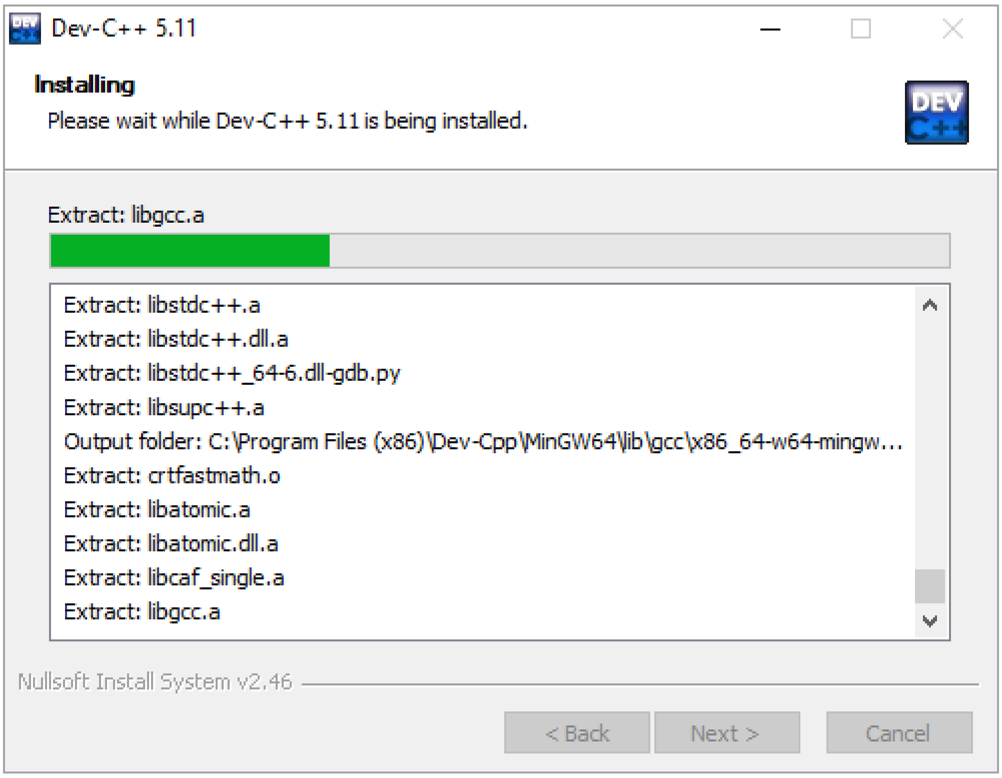
Space required: 346.8MB

Space available: 319.7GB

Nullsoft Install System v2.46

Once we provide the destination folder path, click on Install.

5) The following screenshot shows the progress of the installation.



Installing

Please wait while Dev-C++ 5.11 is being installed.

Extract: libgcc.a

```
Extract: libstdc++.a
Extract: libstdc++.dll.a
Extract: libstdc++_64-6.dll-gdb.py
Extract: libsupc++.a
Output folder: C:\Program Files (x86)\Dev-Cpp\MinGW64\lib\gcc\x86_64-w64-mingw...
Extract: crtfastmath.o
Extract: libatomic.a
Extract: libatomic.dll.a
Extract: libcaf_single.a
Extract: libgcc.a
```

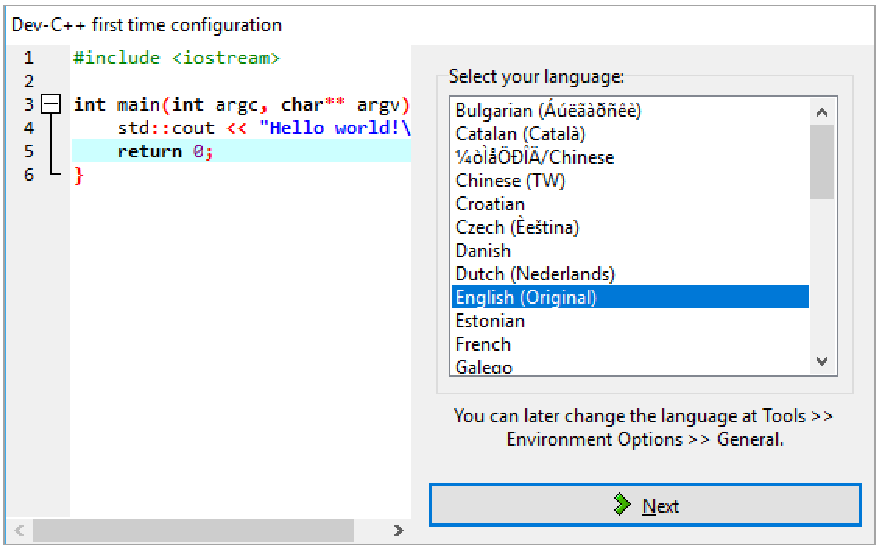
Nullsoft Install System v2.46

Once the installation is over, a “finish” dialog that signals the end of the installation appears. We click finish and then we can launch the dev-C++ IDE.

Now let's see the working of this C++ IDE in detail.

(a) Development Using Dev-C++ IDE

(b) Configuring Dev C++



```
#include <iostream>

int main(int argc, char** argv)
    std::cout << "Hello world!\
    return 0;
}
```

Select your language:

- Bulgarian (Áúëãàðñêè)
- Catalan (Català)
- ¼òÌåÖÐÎÄ/Chinese
- Chinese (TW)
- Croatian
- Czech (Èeština)
- Danish
- Dutch (Nederlands)
- English (Original)
- Estonian
- French
- Galego

You can later change the language at Tools >> Environment Options >> General.

-

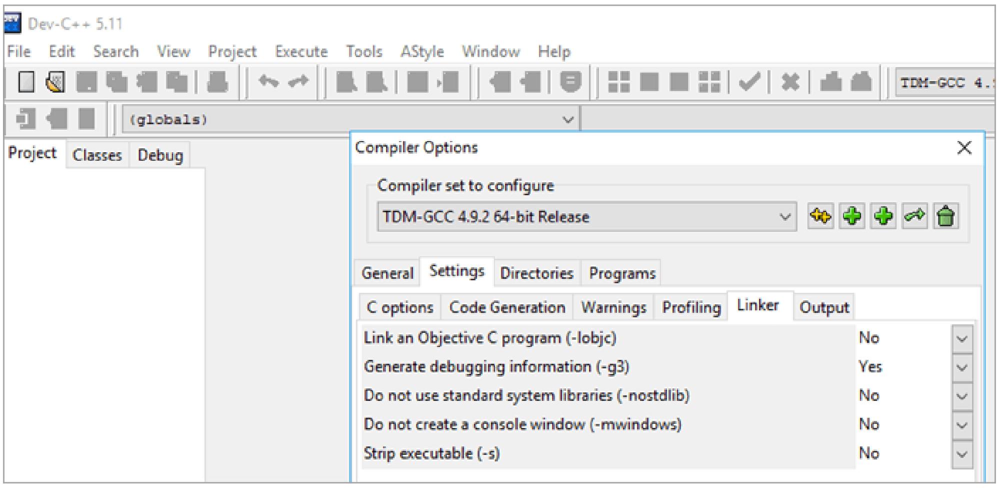
(c) Change Linker Setting For Debugging

After starting the IDE, the first thing we need to ensure is the setting for debugging information to be generated.

Follow the steps below to set the debugging information.

- To change this setting, click on **Tools → Compiler Options.**
- Then click on the "**Settings**" tab on the dialog that pops up.
- Under "**Settings**", we have a "**linker**" tab.
- In the "**linker**" tab there are various options shown. Set "**Yes**" for the option "**Generate Debugging Information (-g3)**".

This is shown in the following screenshot.


Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 4.
(globals)
Project Classes Debug
Compiler Options
Compiler set to configure
General Settings Directories Programs
C options Code Generation Warnings Profiling Linker Output

TDM-GCC 4.9.2 64-bit Release

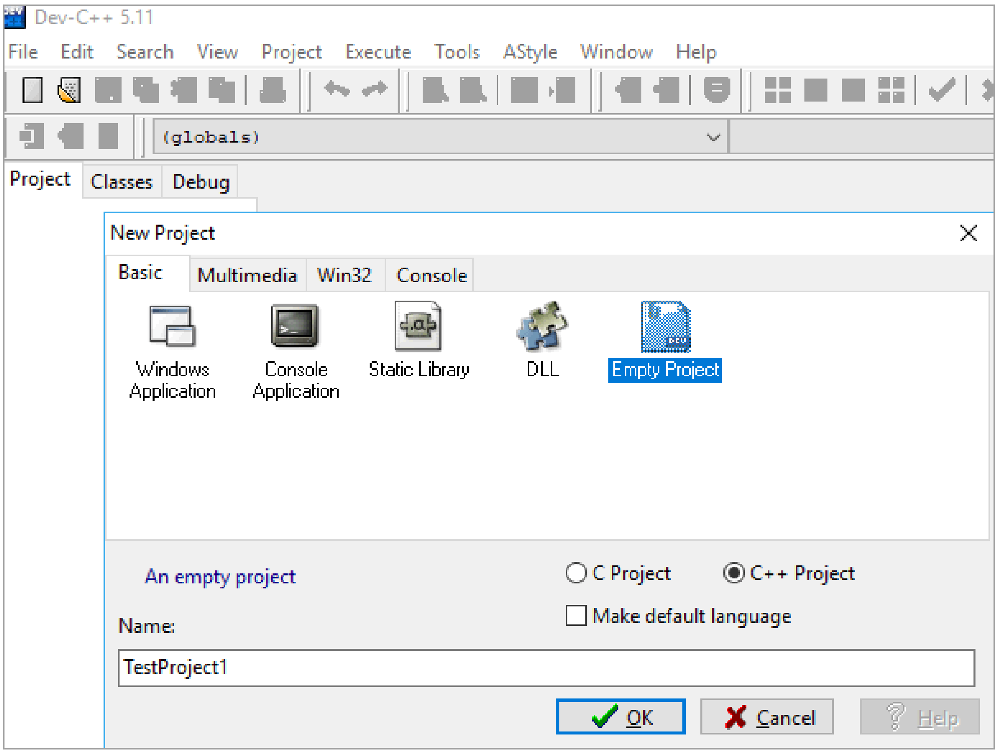
Link an Objective C program (-lobjc)	No
Generate debugging information (-g3)	Yes
Do not use standard system libraries (-nostdlib)	No
Do not create a console window (-mwindows)	No
Strip executable (-s)	No

Click OK, once done.

(d) Create a New Project

To create a new project in dev-C++ we need to follow the below steps:

- Click **File → New → Project.**
- A new dialog opens up as shown below.



An empty project

○ C Project ● C++ Project

□ Make default language

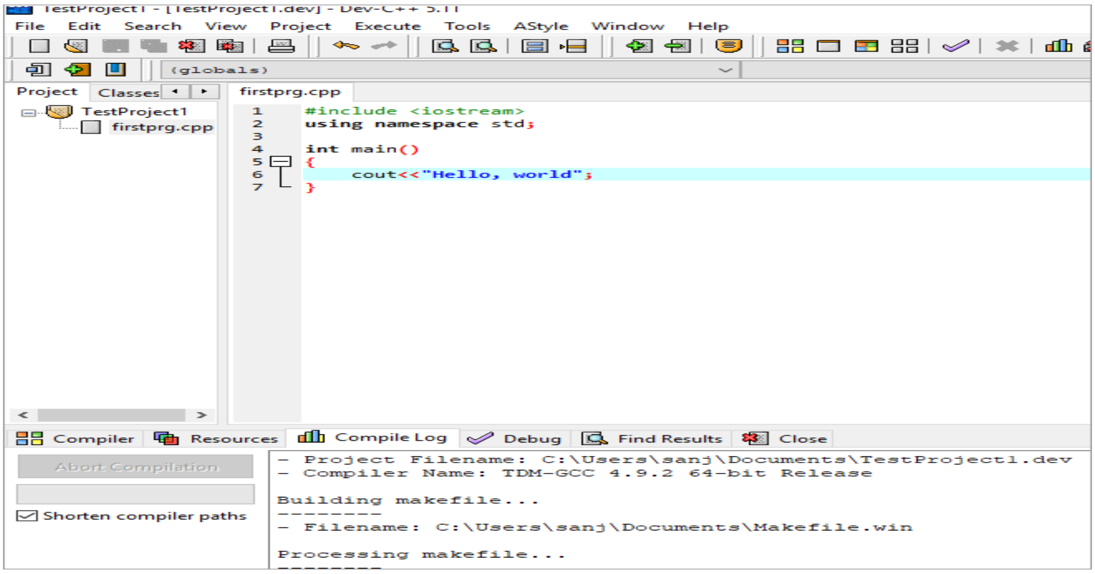
Name: TestProject1

- Here, we can specify the project name. Make sure to select the "Empty Project" and also to check the "C++ Project" button.
- Once the entire information is provided, we can click ok and the IDE will ask for the path where the project is to be saved. When this is done, a workspace will open with the project explorer on the left-hand side that shows the project we just created.
- Now we can add or import the code files into this project.

(e) Add Source File(s)

Adding a file to a project can be done in two ways.

1. Add a new file by clicking **Project →New File** or Right-click on **Project Name** in the project explorer and click **New File**.
2. Another way is to add the existing files to the project. This can be done by clicking **Project →Add to Project** or right-click on **Project Name** in the project explorer and select "**Add to Project…**" This will give a dialog to select files and import them to the project.
3. Once the files are added to the project, the workspace looks as shown below.



TestProject1 - [TestProject1.dev] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
(globals)
Project Classes
TestProject1
firstprg.cpp
Compiler Resources Compile Log Debug Find Results Close
Abort Compilation
☑ Shorten compiler paths

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

int main()
{
    cout<<"Hello, world";
}
```

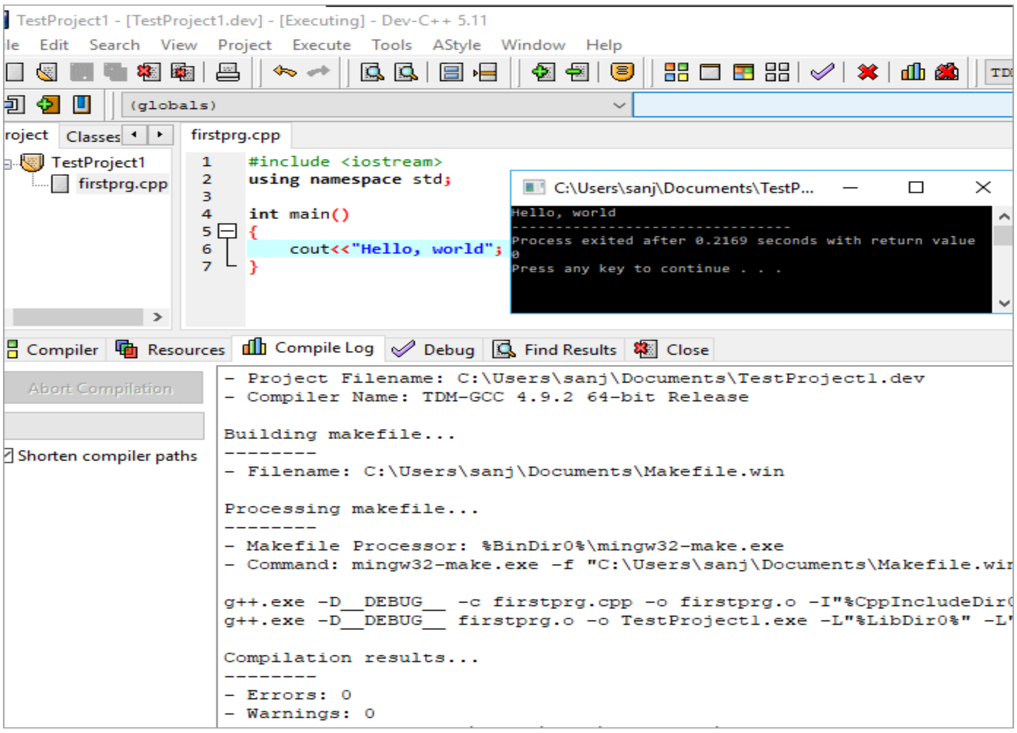
```
- Project Filename: C:\Users\sanj\Documents\TestProject1.dev
- Compiler Name: TDM-GCC 4.9.2 64-bit Release

Building makefile...
--------
- Filename: C:\Users\sanj\Documents\Makefile.win

Processing makefile...
--------
```

(f) Compile/Build & Execute Project

When we have all the code ready for the project, we will now compile and build the project.

Follow the below steps to build and execute the dev C++ project:

- To compile the project, click **Execute → Compile** (or click F9).
- We can see the compilation status in the "**Compile Log**" tab in the workspace.
- If there are any errors whether syntax or linker errors, then they will appear in the compiler tab.
- Once the project is compiled successfully, we need to run it.
- Click on **Execute →Run**.(or click F10)
- The console window that gives us the output will be shown in the below screenshot.



TestProject1 - [TestProject1.dev] - [Executing] - Dev-C++ 5.11
le Edit Search View Project Execute Tools AStyle Window Help
(globals)
roject Classes
TestProject1
firstprg.cpp
C:\Users\sanj\Documents\TestP...
Compiler Resources Compile Log Debug Find Results Close
Abort Compilation
Shorten compiler paths

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

int main()
{
    cout<<"Hello, world";
}
```

```
Hello, world
--------------------------------
Process exited after 0.2169 seconds with return value
0
Press any key to continue . . .
```

```
- Project Filename: C:\Users\sanj\Documents\TestProject1.dev
- Compiler Name: TDM-GCC 4.9.2 64-bit Release

Building makefile...
--------
- Filename: C:\Users\sanj\Documents\Makefile.win

Processing makefile...
--------
- Makefile Processor: %BinDir0%\mingw32-make.exe
- Command: mingw32-make.exe -f "C:\Users\sanj\Documents\Makefile.wi

g++.exe -D__DEBUG__ -c firstprg.cpp -o firstprg.o -I"%CppIncludeDir
g++.exe -D__DEBUG__ firstprg.o -o TestProject1.exe -L"%LibDir0%" -L

Compilation results...
--------
- Errors: 0
- Warnings: 0
```

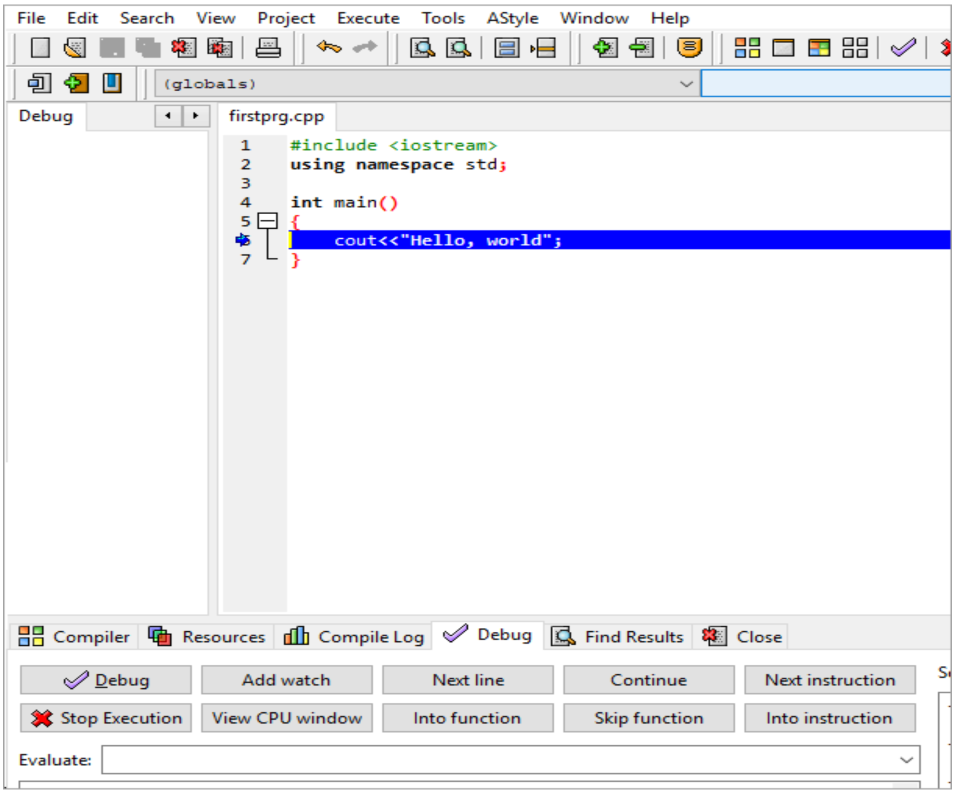
- If there are command line parameters to be passed to the program, we click on **Execute →Parameters**. This will open a dialog using which we can pass parameters.

(g) Debugging In C++ IDE

Sometimes we may not get the desired output from our program although the program is syntactically correct. In such a situation, we can debug the program. The dev-C++ IDE provides the inbuilt debugger.

Follow the below steps to debug the program using Dev-C++ IDE:

- Click **Execute→Debug**. (or click **F5**).
- Once the debug is clicked, we get the debug menu in the IDE, as shown below.



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

int main()
{
    cout<<"Hello, world";
}
```

- Before debugging we can toggle breakpoints using F4 at a particular line of code.
- Using the debug menu, we can use options like add watches, run to cursor, into function, etc. to efficiently debug our program.

Article II. Setting up Code::Blocks on Windows

This tutorial gives you easy-to-follow instructions, with screenshots, for setting up a compiler (the **MINGW compiler**), a tool that will let you turn the code that you write into programs, and **Code::Blocks**, a **free** development environment for C and C++. This tutorial explains how to install Code::Blocks on Windows 2000, XP, Vista or Windows 7.

Section 2.01Step 1: Download Code::Blocks

- Go to this website: http://www.codeblocks.org/downloads
- Follow the link to "Download the binary release" (direct link)
- Go to the Windows 2000 / XP / Vista / 7 section

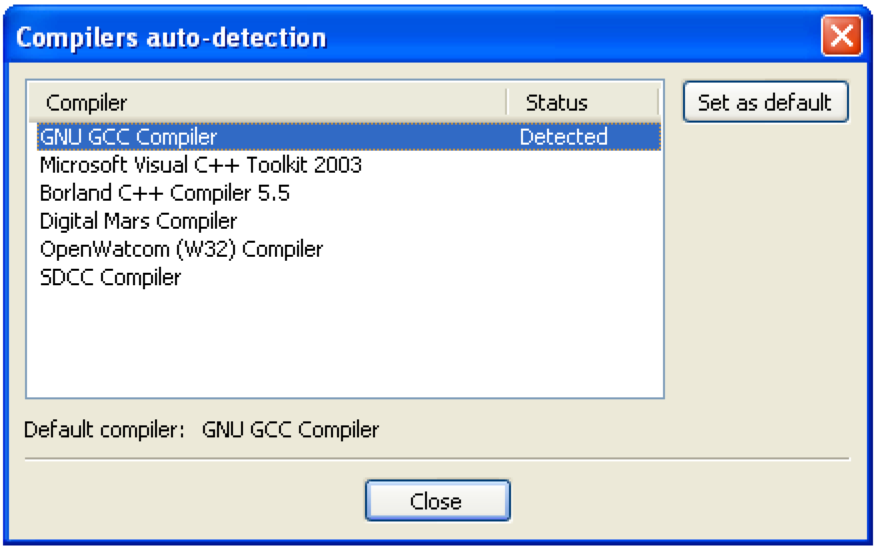
- Look for the file that includes mingw in the name. (The name as of this writing was codeblocks-10.05mingw-setup.exe; the 10.05 may be different).
- Save the file to your desktop. It is roughly 74 megabytes.

Section 2.02Step 2: Install Code::Blocks

- Double click the installer.
- Hit next several times. Other setup tutorials will assume you have installed in **C:\Program Files\CodeBlocks** (the default install location), but you may install elsewhere if you like
- Do a Full Installation
- Launch Code::Blocks

Section 2.03Step 3: Running in Code::Blocks

You will be prompted with a Compilers auto-detection window:



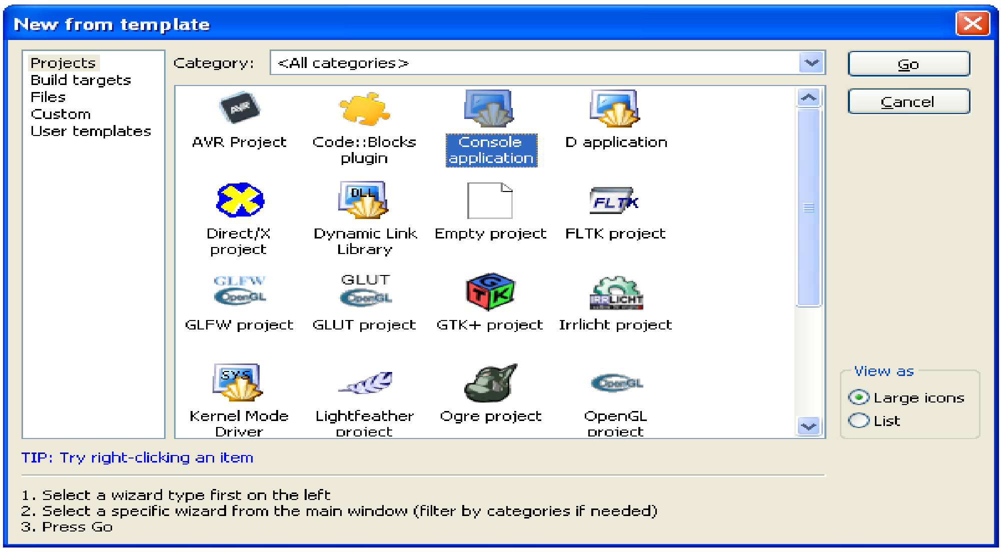
Compilers auto-detection

Compiler	Status
GNU GCC Compiler	Detected
Microsoft Visual C++ Toolkit 2003	
Borland C++ Compiler 5.5	
Digital Mars Compiler	
OpenWatcom (W32) Compiler	
SDCC Compiler	

Default compiler: GNU GCC Compiler

When you get the compiler auto-detection window, just hit OK. Code::Blocks may ask if you want to associate it as the default viewer for C/C++ files--I'd suggest you do. Click on the File menu, and under "New", select "Project..."

The following window will come up:

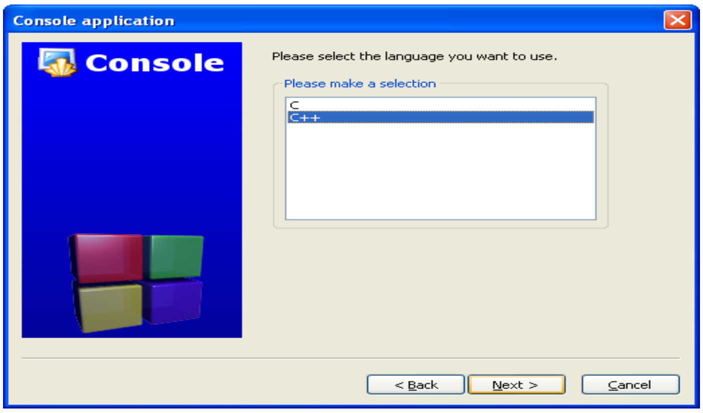


TIP: Try right-clicking an item

1. Select a wizard type first on the left
2. Select a specific wizard from the main window (filter by categories if needed)
3. Press Go

Click on "Console Application" and hit the "Go" button.

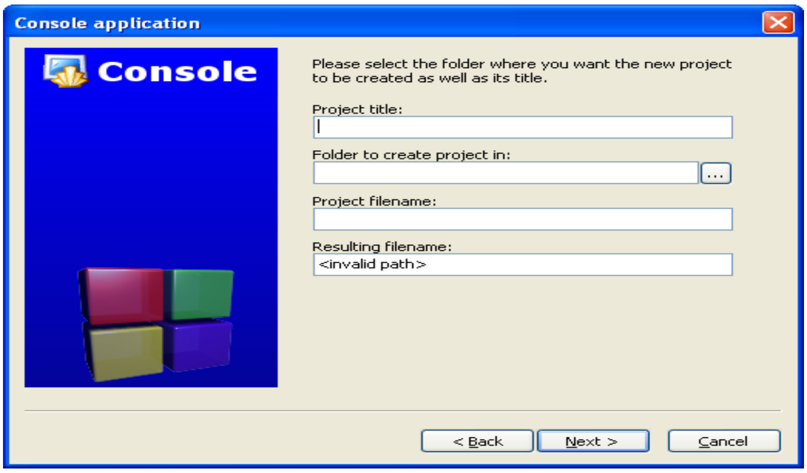
Click next until you get to the Language Selection Dialog:



Please select the language you want to use.

You'll be asked to choose whether you want to use C or C++. If you're not sure, use C++. Otherwise, choose based on the language you are learning.

After clicking "Next", Code::Blocks will then prompt you with where you'd like to save the console application:



Please select the folder where you want the new project to be created as well as its title.

Project title:

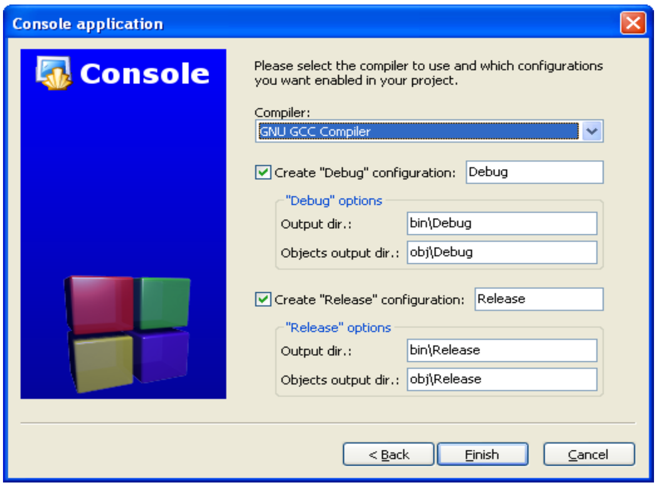
Folder to create project in:

Project filename:

Resulting filename: <invalid path>

I'd recommend you put it in its own folder, as it may create several files (this is especially true if you create other types of projects). You will need to give your project a name, anything will be fine.

Clicking "Next" again will prompt you to set up your compiler:



Please select the compiler to use and which configurations you want enabled in your project.

Compiler: GNU GCC Compiler

☑ Create "Debug" configuration: Debug

"Debug" options

Output dir.: bin\Debug

Objects output dir.: obj\Debug

☑ Create "Release" configuration: Release

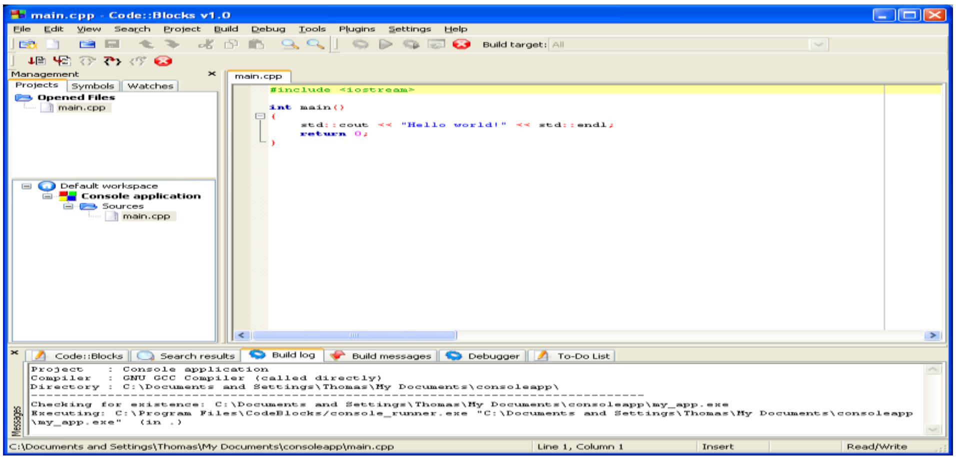
"Release" options

Output dir.: bin\Release

Objects output dir.: obj\Release

You don't need to do anything here. Just accept the defaults by hitting "Finish".

You can now open the main.cpp file on the left:

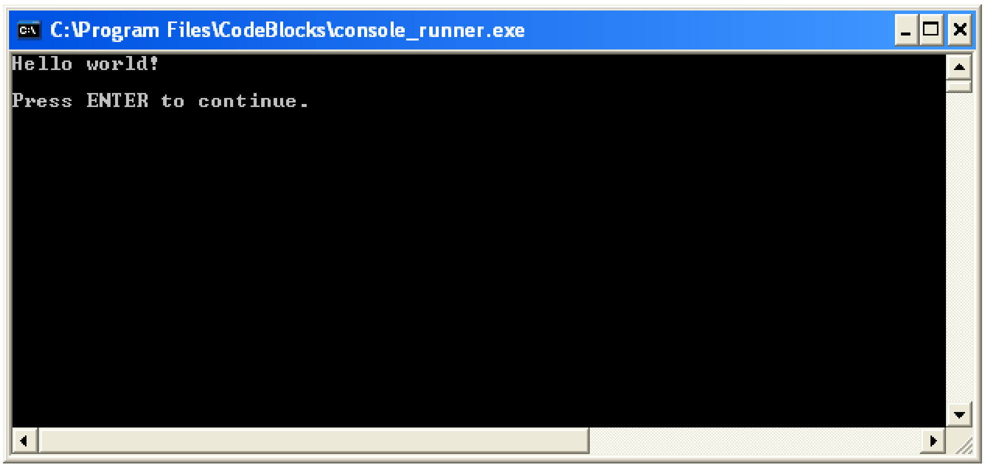

main.cpp - Code::Blocks v1.0
File Edit View Search Project Build Debug Tools Plugins Settings Help
Build target: All
Management
Projects Symbols Watches
Opened Files
main.cpp
Default workspace
Console application
Sources
main.cpp
main.cpp
Code::Blocks Search results Build log Build messages Debugger To-Do List
Messages
C:\Documents and Settings\Thomas\My Documents\consoleapp\main.cpp
Line 1, Column 1
Insert
Read/Write

```
#include <iostream>

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

```
Project    : Console application
Compiler   : GNU GCC Compiler (called directly)
Directory : C:\Documents and Settings\Thomas\My Documents\consoleapp\
--------------------------------------------------------------------------------
Checking for existence: C:\Documents and Settings\Thomas\My Documents\consoleapp\my_app.exe
Executing: C:\Program Files\CodeBlocks/console_runner.exe "C:\Documents and Settings\Thomas\My Documents\consoleapp
\my_app.exe"  (in .)
```

(You may need to expand the contents of the "Sources" folder if you don't see main.cpp.)

At this point, you will have your main.cpp file, which you can modify if you like. For now, it just says "Hello World!", so we can run it as is. Hit F9, which will first compile it and then run it.



```
Hello world!

Press ENTER to continue.
```

You now have a running program! You can simply edit main.cpp and then hit F9 to compile it and run it again.

Linux Installation: Install GCC the C++ compiler on Ubuntu 18.04 Bionic Beaver Linux

We will install the GNU GCC compiler on Linux. To install and work with the GCC compiler on your Linux machine, proceed according to below steps:

- You have to first run the below two commands from your Linux terminal window:

  ```
  sudo apt-get update
  sudo apt-get install GCC
  ```

  This command will install the GCC compiler on your system. You may also run the below command:

  ```
  sudo apt-get install build-essential
  ```

  This command will install all the libraries which are required to compile and run a C++ program.

- After completing the above step, you should check whether the GCC compiler is installed in your system correctly or not. To do this you have to run the below-given command from Linux terminal:

  ```
  g++ --version
  ```

- If you have completed the above two steps without any errors, then your Linux environment is set up and ready to be used to compile C++ programs. In further steps, we will learn how to compile and run a C++ program on Linux using GCC compiler.
- Write your program in a text file and save it with any file name and.CPP extension. We have written a program to display “Hello World” and saved it in a file with the filename “helloworld.cpp” on desktop.

- Now you have to open the Linux terminal and move to the directory where you have saved your file. Then you have to run the below command to compile your file:

```
g++ filename.cpp -o any-name
```

filename.cpp is the name of your source code file. In our case, the name is “helloworld.cpp” and *any-name* can be any name of your choice. This name will be assigned to the executable file which is created by the compiler after compilation. In our case, we choose *any-name* to be “hello”. We will run the above command as:

```
g++ helloworld.cpp -o hello
```

- After executing the above command, you will see a new file is created automatically in the same directory where you have saved the source file and the name of this file is the name you chose as *any-name*. Now to run your program you have to run the below command:

```
./hello
```

This command will run your program in the terminal window.

WEEK #4

OBJECTIVES

- ❒ To help the students in learning the concepts of C++.
- ❒ To help the students in learning about different operators available in C++.
- ❒ To help the students in learning the different decision making statements and control statements used in C++.

OUTCOMES

After completing this, the students would be able to:

- ❒ Understand the concepts of C++.
- ❒ Understand the usage of various operators and their precedence in expression evaluation.
- ❒ Understand the usage of decision making statements and control statements of C++ in real life applications.

PROBLEMS

1# Write a C++ program to check whether a number is even or odd using ternary operator.

2# Write a C++ program to perform the addition of two numbers without using + operator.

3# Write a C++ program to evaluate the arithmetic expression ((a + b / c * d - e) * (f - g)). Read the values a, b, c, d, e, f, g from the standard input device.

4# A Fibonacci sequence is defined as follows: The first and second terms in the sequence are 0 and 1. Subsequent terms are found by adding the preceding two terms in the sequence. Write a C++ program to generate the first n terms of the sequence.

5# Write a C++ program to generate all the prime numbers between 1 and n, where n is a value supplied by the user.

6# A character is entered through keyboard. Write a C++ program to determine whether the character entered is a capital letter, a small case letter, a digit or a special symbol using if-else and switch case. The following table shows the range of ASCII values for various characters.

Characters ASCII values A – Z: 65 – 90, a – z: 97 – 122, 0 – 9: 48 – 57

Special symbols 0 – 47, 58 – 64, 91 – 96, 123 – 127

7# Write a C++ program to find the roots of a quadratic equation.

8# Write a C++ program to check whether a given 3 digit number is Armstrong number or not.

WEEK #5

OBJECTIVES

- ❒ To help the students in learning the concepts of *arrays* (1D and 2D) in C++.
- ❒ To help the students in implementing and handling 1D and 2D arrays with the help on various examples.
- ❒ To help the students in understanding various operations associated with the arrays.

OUTCOMES

After completing this, the students would be able to:

- ❒ Handle arrays (1D and 2D) in the programs.
- ❒ Develop & implement simple real life examples of arrays.

PROBLEMS

1# Write a C++ program to find the second largest integer in a list of integers.

2# Write a C++ program to perform the following:

- Addition of two matrices
- Multiplication of two matrices

3# Write a C++ program to count and display positive, negative, odd and even numbers in an array.

4# Write a C++ program to merge two sorted arrays into another array in a sorted order.

5# Write a C++ program to find the frequency of a particular number in a list of integers.

WEEK #6

OBJECTIVES

- To help the students in learning the concepts of *pointers* in C++.
- To help the students in implementing and handling pointers with the help on various examples.
- To help the students in understanding various operations associated with the pointers.

OUTCOMES

After completing this, the students would be able to:

- Handle pointers in the programs.
- Develop & implement simple examples of pointers.
- Implement extensive use of pointers for memory, array, structures and functions.

PROBLEMS

1# Write a C++ program to concatenate two strings using pointers.

2# Write a program for reading elements using pointer into array and display the values using array.

i. Declare set of elements.

ii. Declare pointer and initialize it to first element address of set of elements (array).

iii. Repeat loop until pointer reaching to last element and display each element.

3# Write a program through pointer variable to sum of n elements from array.

4# Write a program for reading elements using pointer into array and display the values using array.

5# Write a C++ program to reverse a string using pointers.

WEEK #7

OBJECTIVES

- ❒ To help the students in implementing and handling pointers with the help on various examples.
- ❒ To help the students in understanding various operations associated with the pointers.
- ❒ To help the students in understanding the concept of *this* pointer

OUTCOMES

After completing this, the students would be able to:

- ❒ Handle pointers in the programs.
- ❒ Develop & implement simple examples of pointers.
- ❒ Implement extensive use of pointers for memory, array, structures and functions.

PROBLEMS

1# Write a C++ program using pointers to compute the sum, mean and standard deviation of all elements stored in an array of **n** real numbers.

2# Write a C++ program to create three objects for a class named pntr_obj with data members such as roll_no & name. Create a member function set_data() for setting the data values and print() member function to print which object has invoked it using *'this'* pointer

3# Develop a C++ program to find the greatest of two numbers using this pointer which returns the member variable.

4# Write a C++ program to implement flight class with data member as flight_no., source destination and fare. Write a member function to display the flight information using this pointer.

5# Write a C++ program to use *this* pointer and return pointer reference.

WEEK #8

OBJECTIVES

❒ To help the students in learning the concepts of *strings* in C++.

❒ To help the students in implementing and handling strings operations on string with the help on various examples.

OUTCOMES

After completing this, the students would be able to:

❒ Handle strings manipulation in the programs.

❒ Develop & implement simple real life examples of strings/string functions.

PROBLEMS

1# Write a C++ program that uses functions to perform the following operations:

i. To insert a sub string into a given main string from a given position.

ii. To delete n characters from a given position in a given string.

2# Write a C++ program to determine if the given string is a palindrome or not.

3# Write a C++ program to find a string within a sentence and replace it with another string.

4# Write a C++ program that reads a line of text and counts all occurrence of a particular word.

5# Write a C++ program that displays the position or index in the string S where the string T begins, or 1 if S doesn't contain T.

WEEK #9

OBJECTIVES

- ❒ To help the students in implementing and handling about writing your own functions with the help on various examples.
- ❒ To help the students in learning Recursion.
- ❒ To help the students in understanding various operations associated with the function using array.
- ❒ To learn how to write queries for creating tables with constraints and inserting and retrieving the records.

OUTCOMES

After completing this, the students would be able to:

- ❒ Understand how to access and use library functions in the programs.
- ❒ Understand Functions & Recursion.
- ❒ Understand the different constraints that can be applied on a table.
- ❒ Write DML, DDL based queries.

PROBLEMS

1# Write C programs that use both recursive and non-recursive functions to find:

a) The factorial of a given integer.

b) To find the greatest common divisor of two given integers.

2# Write C programs that use both recursive and non-recursive functions to solve towers of Hanoi problem.

3# Write a C++ program to print the transpose of a given matrix using function.

4# Write a C++ program to swap two number by both call by value and call by reference mechanism, using two functions swap_value() and swap_reference respectively, by getting the choice from the user and executing the user’s choice by switch-case.

5# Write a C++ program to display all array elements using recursion.

6# Write a C++ program to find sum of elements of array using recursion.

7# Write a C++ program to find maximum and minimum elements in array using recursion.

8# Consider the insurance database given below. The primary keys are made bold and the data types are specified.

PERSON(driver_id:string , name:string , address:string)

CAR(regno:string , model:string , year:int)

ACCIDENT(report_number:int , accd_date:date , location:string)

OWNS(driver_id:string , regno:string)

PARTICIPATED(driver_id:string , regno:string , report_number:int , damage_amount:int).

- ❒ Create the above tables by properly specifying the primary keys and foreign keys.
- ❒ Enter at least five tuples for each relation.
- ❒ Update the damage amount for the car with specific regno in the accident with report number 12 to 25000.
- ❒ Add a new accident to the database.
- ❒ Find the total number of people who owned cars that were involved in accidents in the year 2008.
- ❒ Find the number of accidents in which cars belonging to a specific model were involved.

WEEK #10

OBJECTIVES

❒ To help the students in learning the concepts of *Structure* in C++.

❒ To help the students in implementing *structure* with the help on various examples.

❒ To learn how to write queries for creating tables with constraints and inserting and retrieving the records.

OUTCOMES

After completing this, the students would be able to:

❒ Handle problem based on *structure*.

❒ Develop & implement simple real life examples of structure.

❒ Understand the different constraints that can be applied on a table.

PROBLEMS

1# Write a C++ program that uses functions to perform the following operations:

i. Reading a complex number

ii. Writing a complex number

iii. Addition and subtraction of two complex numbers

iv. Multiplication of two complex numbers. Note: represent complex number using a structure.

2# Write a C++ program to compute the monthly pay of 100 employees using each employee's name, basic pay. The DA is computed as 52% of the basic pay. Gross-salary (basic pay + DA). Print the employees name and gross salary.

3# Create a Book structure containing book_id, title, author name and price. Write a C++ program to pass a structure as a function argument and print the book details.

4# Create a union containing 6 strings: name, home_address, hostel_address, city, state and zip. Write a C++ program to display your present address.

5# Write a C++ program to define a structure named DOB, which contains name, day, month and year. Using the concept of nested structures display your name and date of birth.

6# Consider the following database for a banking enterprise. BRANCH(branch_name:string , branch_city:string , assets:real)

ACCOUNT(accno:int , branch_name:string , balance:real)

DEPOSITOR(customer_name:string , accno:int)

CUSTOMER(customer_name:string, customer_street:string , customer_city: string)

LOAN(loan_number:int , branch_name:string , amount:real)

BORROWER(customer_name:string , loan_number:int)

a) Create the above tables by properly specifying the primary keys and foreign keys.

b) Enter at least five tuples for each relation.

c) Find all the customers who have at least two accounts at the main branch.

d) Find all the customers who have an account at all the branches located in a specific city.

e) Demonstrate how you delete all account tuples at every branch located in a specific city.

WEEK #11

OBJECTIVES

- ❒ To help the students in learning the concept of OOPs.
- ❒ To help the students how to write a sample program using OOPs.
- ❒ To help the students in understanding the use of class and objects.
- ❒ To help the students in understanding the concept of Access Specifier.

OUTCOMES

After completing this, the students would be able to:

- ❒ Write, debug and run a sample program using OOPs.
- ❒ Create simple real life applications.
- ❒ Understand following points:
  - ✓ Class and object.
  - ✓ Crating class and object.
  - ✓ Defining function in different ways.
- ❒ Concept of Access Specifier like private, public etc.

PROBLEMS

1# Write a program in C++ to display your name, Branch, Year on to the computer screen without using classes and object. All information should be displayed in the separate line.

2# Write a menu driven program in C++ to perform all basic arithmetic operation addition, subtraction, multiplication, and division of two given values. Program receives two values and required operation to be performed from the keyboard and display particular result of the required operation.

3# Write a menu driven program in C++ that receives 4 digit integer value the keyboard and perform following operations:

i) Reverse of that no.

ii) sum of number with its reverse.

iii) sum of alternative digits(1 digit+3 digit and 2 digit+4 digit)

4# Write a menu driven program in C++ to receive integer number and convert equivalent binary, octal, hexadecimal number.

5# Write a menu driven program in C++ to perform all basic arithmetic operation addition, subtraction, multiplication, and division of two given values using function and switch case. Program receives two values and required operation to be performed from the keyboard and display particular result of the required operation.

6# Define a class Bank Account to represent a bank account. Include the following members:

Data Members:

- Name of the depositor
- Account Number
- Type of account
- Balance amount in the account

Member Functions:

- To assign initial value
- To deposit an amount
- To withdraw an amount after checking

WEEK #12

OBJECTIVES

❒ To help the students in understanding the concept of Constructor.

❒ To help the students in understanding the need for constructors and destructors in their programs.

❒ To help the students in understanding the use of copy constructor and dynamic constructors.

❒ To learn how to write queries for creating tables with constraints and inserting and retrieving the records.

OUTCOMES

After completing this, the students would be able to:

❒ Understand the use of copy constructor and dynamic constructor.

❒ Create a simple real life applications based on the constructors and destructors with static members.

❒ Understand the different constraints that can be applied on a table.

PROBLEMS

1# Write a program in C++ to demonstrate default constructor. Create a class having two data members in the private section. Define a default constructor to initialize these data members to initial value and display these values with the help of member function.

2# Write a program in C++ to demonstrate parameterized/constructor overloading constructor. Create a class calculator that contains four data members in it. Initialize data members with different values using parameterized constructor and perform various arithmetic operation over these values and display result on to the computer screen.

3# Create a class called Triangle that stores the length of the base and height of a right triangle in two private instance variables. Include a constructor that sets these values. Define two functions. The first is hypo( ), which returns the length of the hypotenuse. The second is area ( ), which returns the area of the triangle.

4# Create a class for counting the number of objects created and destroyed within various block using constructor and destructors.

5# Create an inter University Database with the following relations. Include at least four attributes for each table.

- University
- Department
- Program
- Course
- Syllabus
- Faculty(Teacher)

a) Create above tables and mention primary keys and foreign keys. Also create secondary index for each table.

b) Insert at least 5 relevant records in each of the created tables.

c) Write following SQL queries based on above created database:

i) List of Universities situated at Delhi.

ii) List of all Departments of AMU.

iii) Find the location of JNU.

iv) List of all Programs run by University of Jammu.

v) List of Universities that run Program "MCA".

vi) List of Courses of "MCA" run by AMU.

vii) List of Faculties specialized in "Information Security" across different universities.

viii) Syllabus of "Computer Architecture" of different Universities.

WEEK #13

OBJECTIVES

- ❒ To help the students in learning operator overloading and the rules for overloading operators in C++.
- ❒ To help the students in understanding the use of overloading of new and delete operators.
- ❒ To help the students in leaning the concept of Inheritance.
- ❒ To help the students in understanding the use of different forms of inheritance in their programs.

OUTCOMES

After completing this, the students would be able to:

- ❒ Use of binary operator overloading and understand the use of non-member function in overloading.
- ❒ Create a simple real life applications based on Type conversion i.e. type conversion - basic type to class type, class type to basic type, class type to another class type.
- ❒ Know the working/avoid of ambiguity in multiple and multi-path inheritance.
- ❒ Create a simple real life applications based on virtual base class and overriding member functions.

PROBLEMS

1# Declare a class Number that contains two data member value1 and value2 of the type of integer, define constructor to give initial value, and perform addition ,subtraction, multiplication and division of these two numbers using operating overloading of +,-,*,/ operator respectively [hint- binary operator overloading using member function]

2# Declare a class Number1 that contains two data member value1 and value2 of the type of integer, define constructor to give initial value, and perform addition, subtraction, multiplication and division of these two numbers using operating overloading of +, -, *, / operator respectively [hint- binary operator overloading using friend function]

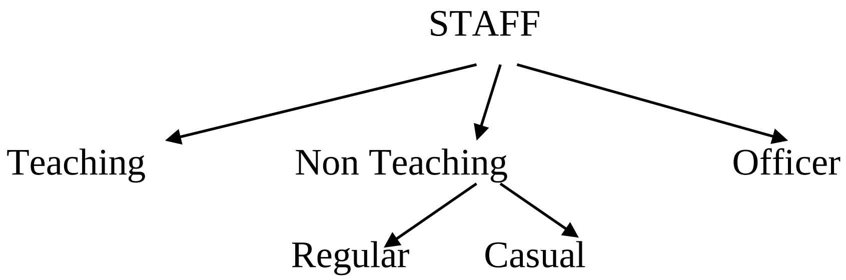
3# Declare a class Number3 that contains a data member value of the type of integer, define constructor to give initial value, and perform unary minus, increment and decrement this number using operating overloading of -, ++, -- operator respectively [hint- Unary operator overloading using member function]

4# Write a program to demonstrate explicit type conversion

- from basic type to user defined data type.
- from User Defined data type to Basic data type data type.

5# Create a class publication which has title of book and writers name. Create other class sales which account no. of sales for every month (upto 3 months) and then calculate total sales.

6# Write a program to demonstrate the following:



7# Write a program to solve the ambiguity problem in inheritance where two different classes are inherited from single base class and a new class is derived from these two derived classes. How this problem is solved with the help of virtual base class concept.

WEEK #14

OBJECTIVES

- ❒ To help the students in learning the concept of binding in C++.
- ❒ To help the students how to write a sample program based on early binding and late binding in C++.
- ❒ To help the students in understanding the use of binding, virtual functions in their programs.
- ❒ To help the students in understanding the use of polymorphism.
- ❒ To help the students in learning the concept of streams in C++.
- ❒ To help the students in understanding the use of hierarchy of file stream classes in their programs.
- ❒ To help the students in understanding the use of Exception handling mechanism in their programs.
- ❒ To help the students in understanding the use of throwing mechanism and catching mechanism

OUTCOMES

After completing this, the students would be able to:

- ❒ Write, debug and run a sample program in C++.
- ❒ Differentiate between virtual functions and pure virtual functions.
- ❒ Create a simple real life applications based on the polymorphism.
- ❒ Differentiate the way of reading/writing of files, accessing records randomly, updating files.
- ❒ Create a simple real life applications based on the file handling.
- ❒ Understand hierarchical error handling: **specific errors** (think: **derived classes**) can be handled closer to where the error occurred; for instance deep in the internals of some library. More general errors (think: base classes) can

be handled higher up in the hierarchy; for instance, in client code that uses the library in C++.

- ❒ Differentiate try-catch blocks and throw.
- ❒ Create a simple real life applications based on the exceptional Handling statements.

PROBLEMS

1# Write a program to use 'this' pointer to find elder from two person. Define a class Person to store age of the person. Define constructor/member function to give initial value to the data member age. And then define a function elder to compare ages of two different persons using this pointer to find out the elder person.

2# Create a simple "shape" hierarchy: a base class called Shape and derived classes called Circle, Square, and Triangle. In the base class, make a virtual function called draw ( ), and override this in the derived classes. Make an array of pointers to Shape objects that you create on the heap (and thus perform up casting of the pointers), and call draw ( ) through the base-class pointers, to verify the behavior of the virtual function. If your debugger supports it, single-step through the code.

3# Write a small program to show the difference between calling a virtual function inside a normal member function and calling a virtual function inside a constructor. The program should prove that the two calls produce different results.

4# Write a program in C++ to calculate mean value of n numbers using friend function.

5# Write a program to accept five different numbers by creating a class called friend func1 and friend func2 taking 2 and 3 arguments respectively and calculate the average of these numbers by passing object of the class to friend function.

6# Write a program in C++ to display student’s information using friend function.

7# Write a C++ program to write text in the file. Read the text from the file from end of file. Display the contents of the file in reverse order.

8# Write a C++ program to count the no. of characters present in the file.

9# Create a class with a main( ) that throws an object of class Exception inside a try block. Give the constructor for Exception a String argument. Catch the exception inside a catch clause and print the String argument. Add a finally clause and print a message to prove you were there.

CREDITS

The following Lab Manual Up-gradation Committee updated the Lab Manual:

➪ **Prof. Aasim Zafar (Chairperson)**
➪ Mr. Suhel Mustajab
➪ Dr. Arman Rasool Faridi
➪ Dr. Faisal Anwer
➪ Dr. Mohammad Nadeem

The following committee members originally designed the Lab Manual:

➪ Prof. Mohammad UbaidullahBokhari
➪ Dr. Arman Rasool Faridi
➪ Dr. Faisal Anwer
➪ Prof. Aasim Zafar (Convener)

Design & Compilation:

➪Mr. Sunil Kumar Sharma

Revised Edition: **July, 2021**

Department of Computer Science
A.M.U., Aligarh, (U.P.), India