

SECTION 1 DBMS LAB

Structure	Page No.
1.0 Introduction	05
1.1 Objectives	05
1.2 Introduction to MS-Access	05
1.3 Database Creation	13
1.4 Use of DBMS Tools/ Client-Server Mode	15
1.5 Forms and Procedures	17
1.6 Summary	20
1.7 Further Readings	20

1.0 INTRODUCTION

By now, you must have obtained the practical skills of several programming languages. However, when we want to create a secure, managed database application, we need not re-start from scratch and develop a huge system using a programming language; rather we use a database management system: application software. This software allows us to create database, query, report and many more operations with database. This section attempts to provide you the basic skills of data organisation, including database creation, integrity enforcement, query formulation, forms and report creation, etc. You should write SQL queries as well as work using interface provided in software packages. For the present practical we have selected MS-Access. However, you must try to develop some applications using MySQL.

You must go through the MCS-023 courseware in order to get the best of those sessions. During the practical sessions you can make suitable assumptions if necessary.

1.1 OBJECTIVES

By the end of the practical sessions of this section, you will be able to:

- create databases using a user interface and SQL command;
- create integrity and constraints on databases;
- develop forms/reports using sample interface;
- write SQL queries; and
- provide a practical overview of advanced concepts like triggers, assertion, views, etc.

1.2 INTRODUCTION TO MS-ACCESS

This topic gives you an introduction to MS-Access and the basic components of MS-Access will also be discussed in this section. But before we look at the Access software and its capabilities, let us recollect what databases are, just go back to your school days, when you used to maintain different copies of your 'Home Work Assignment' and 'Class Assignment'. In those copies on the first page you used to make the 'Index', which contained the headings as Serial no., Chapter, Date, and Remarks. And under those headings, the details of all the 'Assignments' we used to store. Why did we store these details? What was that? Was it a database? Index! You mean to say that 'Index' was a database? YES. *A database is a collection of data related to a particular topic.* Employee records in a file cabinet, a stamp collection in an album – each of these collections is nothing but a database. Database, typically

consists of a heading that describes the type of information it contains, and each row contains some information. In database terminology, the columns are called fields and the rows are called records. This kind of organization in a database is called a table. A database management system (DBMS) is a system that stores and retrieves information in a database. Data management involves creating, modifying, deleting and adding data in files and using this data to generate reports or answer ad-hoc queries. The software that allows us to perform these functions easily is called a DBMS.

Microsoft Corporation introduced a Relational Database management system for the windows platform in 1992 called MS-Access. Microsoft Access is a development environment used to create computer databases.

Start the MS-Access

For starting MS-Access you must have a licensed copy of it, which is available along with MS-Office Professional.

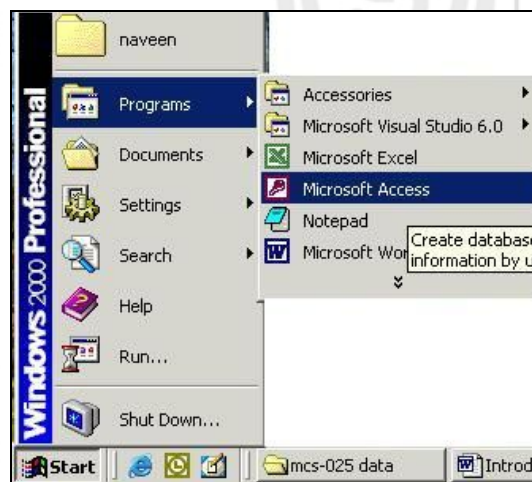


Figure 1: Starting MS-Access

After opening Access as indicated in *Figure 1* above, you will be presented with the Window shown in Figure 2. You can select one of the first two options if you are creating a new database, then go to the second option. If you want to edit an existing database, then go to the third option as shown in *Figure 2*.

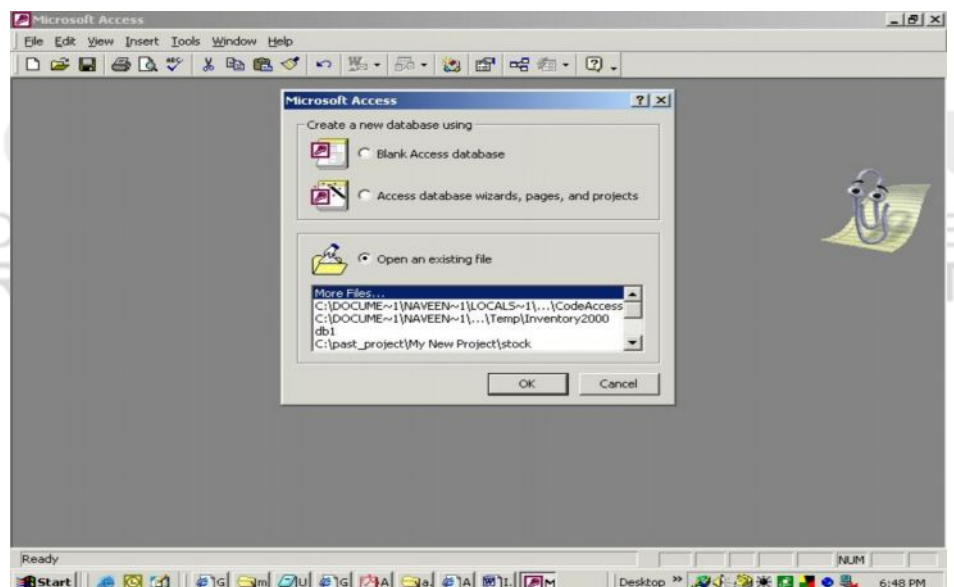


Figure 2: Starting an existing file or creating a new database

Open an existing database

If the database was opened recently on the computer, it will be listed on the main window (as shown in *Figure 2*). Highlight the database name and click OK.

To create a new database

Unlike other office software, you must save an Access database before you start working on it. After selecting “Blank Access database”, you will first be prompted to specify a location and name for the database.

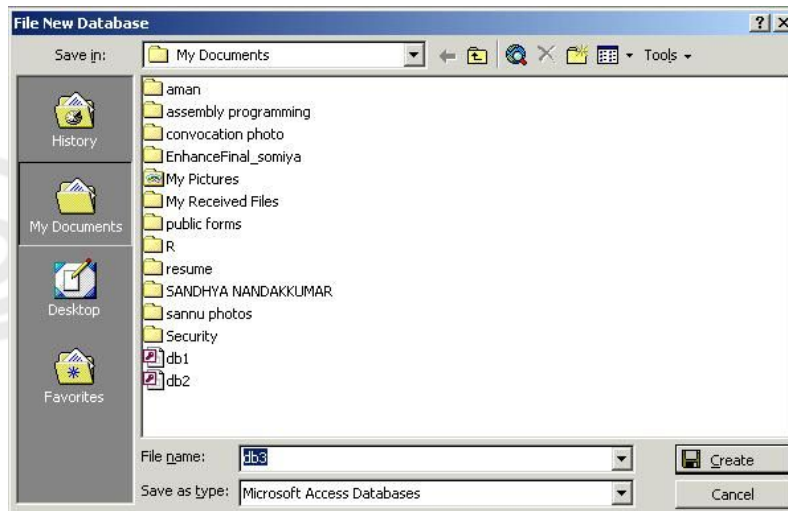


Figure 3: A sample Database Save Screen

You can select the folder where your database should reside and type the name of the database in the **File name** and click the **Create** button.

Database Components

The Database Window as shown below in *Figure 4* organizes all of main objects in the database like tables, queries, form and reports. Further in this we will discuss all these important components of database, which you will need in your lab exercises.

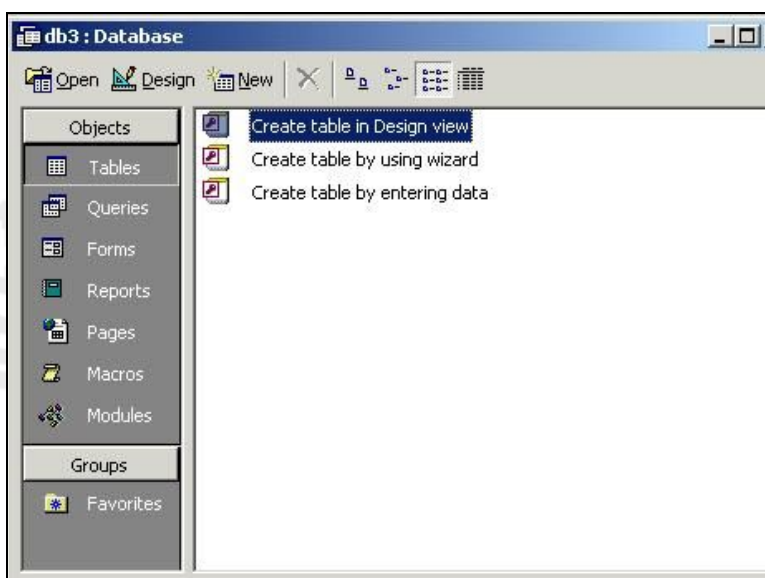


Figure 4: Database Components

Introduction to Tables

A Microsoft Access database is a collection of database files, which are also known as Tables. And each database (a table) is a collection of records, and a record is a collection of fields. You can also understand that the tables are a collection of cells that store information similar to the way an MS-Excel (If you don't know about Excel you can go and check it) worksheet does. MS-Access provides three ways to create a table.

1. Create table in Design view will allow you to create the fields of the table. (Design view is the best way for you).
2. Create table using wizard. (This is best when you are beginning to learn).
3. Create table by entering data, will give you a blank datasheet with unlabelled columns that looks much like an Excel worksheet.

Let us introduce you to Soft Garments, wholesalers for shirts, trousers, and T-shirts. They purchase from various manufacturers and wholesalers. The company has four departments – Sales, Accounts, Stores and Payroll. There are around 2000 employees working under the organization. The company wants to maintain a database, which will store the details and the entire information about all the employees. They want to store the Employee Code, Employee Name, Date of Birth, Date of Joining, Designation, Department and Photographs of the Employees.

Now, if the *Soft Garments* wants to store the employee details, they will have to make a table, which will be a part of some database. The information about one employee will make one record of that table, and the information will be stored under fields as shown in *Figure 5*, fields are *EmployeeID* and *FirstName* and others.

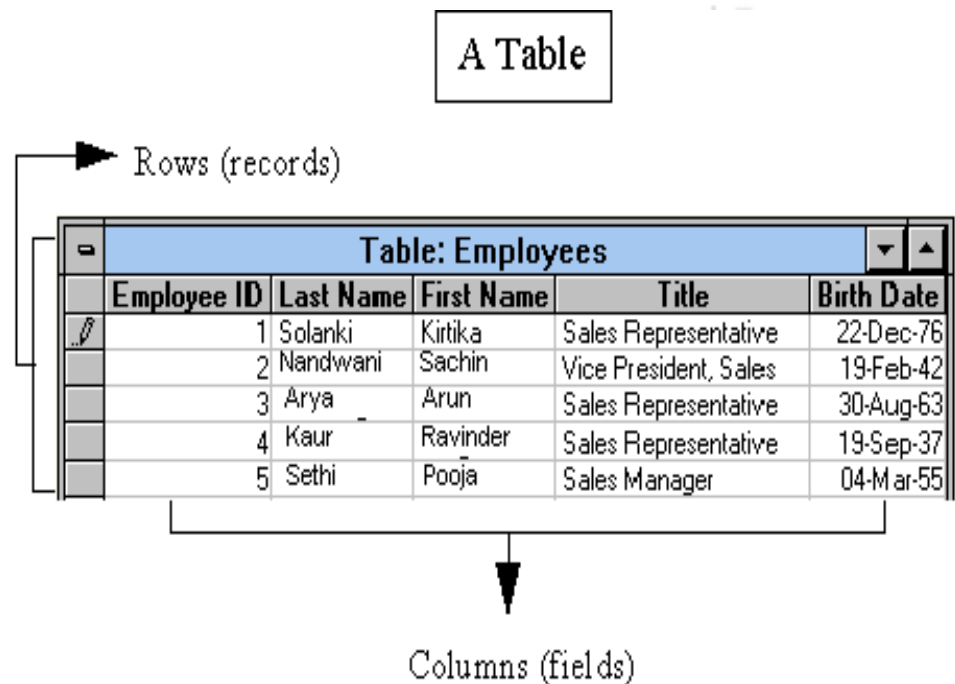


Figure 5: Records and fields of an Employees Table

Each record in a table contains the same set of fields and each field contains the same type of information for each record.

Introduction to Queries

Queries select records from one or more tables in a database so they can be viewed, analyzed, and sorted on a common datasheet. The resulting collection of records, called a dynaset (short for dynamic subset), is saved as a database object and can

therefore be easily used in the future. The query will be updated whenever the original tables are updated. Types of queries are select queries that extract data from tables based on specified values, find duplicate queries that display records with duplicate values for one or more of the specified fields, and find unmatched queries display records from one table that do not have corresponding values in a second table.

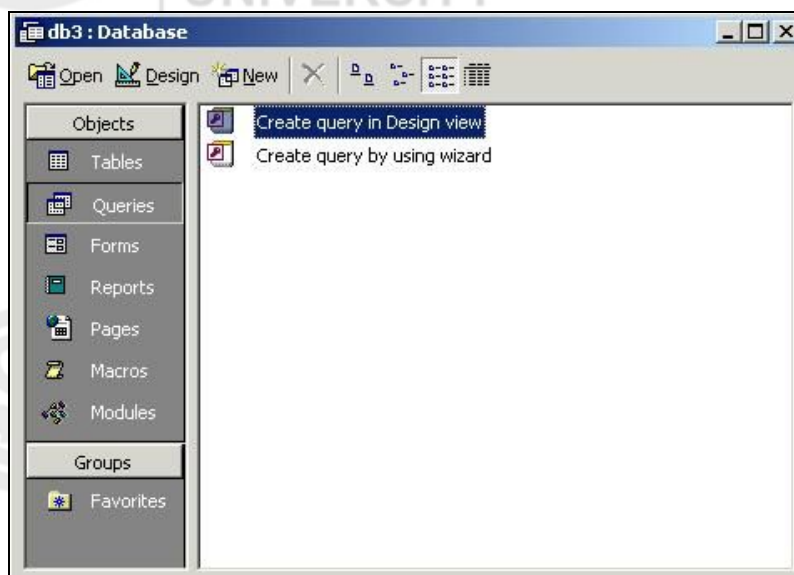


Figure 6: Creating Queries

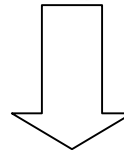
Assume that you are a senior executive in the *Soft Garments* and heading the payroll department. One day the manager of the company calls you, and wants to know how many employees are in 'A' grade. Will you be in a position to answer that Query, right at that moment? May be Yes, May be No. Keeping track of 2000 employees is quite difficult. Not to worry. The manager had a query, he asked you. If you don't know the answer, since you kept your data in database, you can ask the 'Query' to your database.

In MS-Access, A Query is a question you ask about the data in your database. The data that answers the question can be from a single table or several – the query brings the information together.

For solving the above query asked by the manager you can write the following query in access *SQL view*. As shown in *Figure 7* after performing this query on the Employee table you will get the result showing details about employees are who in Grade A. In this example, you have very few employees listed but it is really helpful when the number of employees is huge like 2000 or 20000.

```
SELECT [Employees].[Grade], [Employees].[EmployeeID],
[Employees].[LastName], [Employees].[FirstName], [Employees].[Title]
FROM Employees
WHERE ((([Employees].[Grade])="A"));
```


Employee ID	Last Name	First Name	Title	Grade
1	Solanki	Kiritika	Sales Representative	A
2	Naveen	Kumar	Sales Executive	A
3	Akshay	Kumar	Sales Manager	B
4	Shasi	Bhushan	Vice President, Sales	C



Grade	Employee ID	Last Name	First Name	Title
A	1	Solanki	Kiritika	Sales Representative
A	2	Naveen	Kumar	Sales Executive

Figure 7: Result of query performed on an Employee table

Forms and Reports

Forms are used as an alternative way to enter data into a database table. There are two ways in which you can view the data, stored in a table. Those ways are:

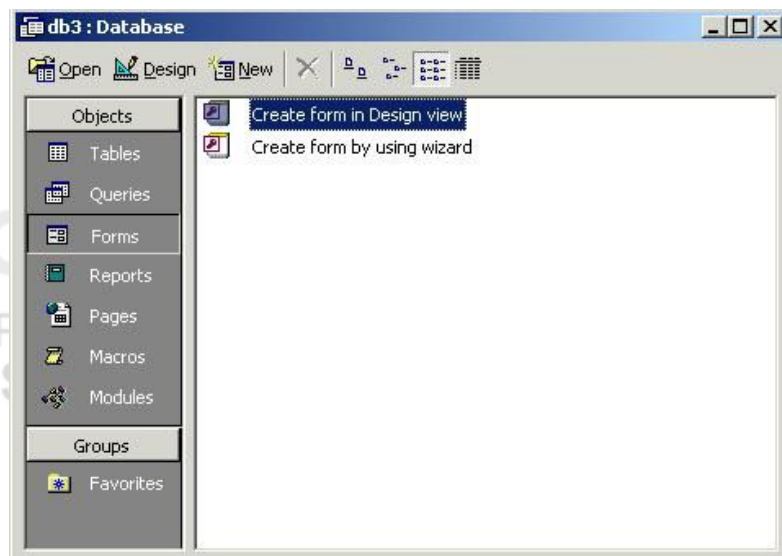


Figure 8: Creating Forms

To Create a Form Using Wizard

To create a form using the assistance of the wizard, follow these steps:

Click the Create form by using wizard option on the database window. From the Tables/Queries drop-down menu, select the table or query whose datasheet the form will modify. Then, select the fields that will be included on the form by highlighting each one, the Available Fields window and clicking the single right arrow button > to move the field to the Selected Fields window as shown in *Figure 9*. To move all of the fields to Select Fields, click the double right arrow button >>. After the proper fields have been selected, click the Next button to move on to the next screen.

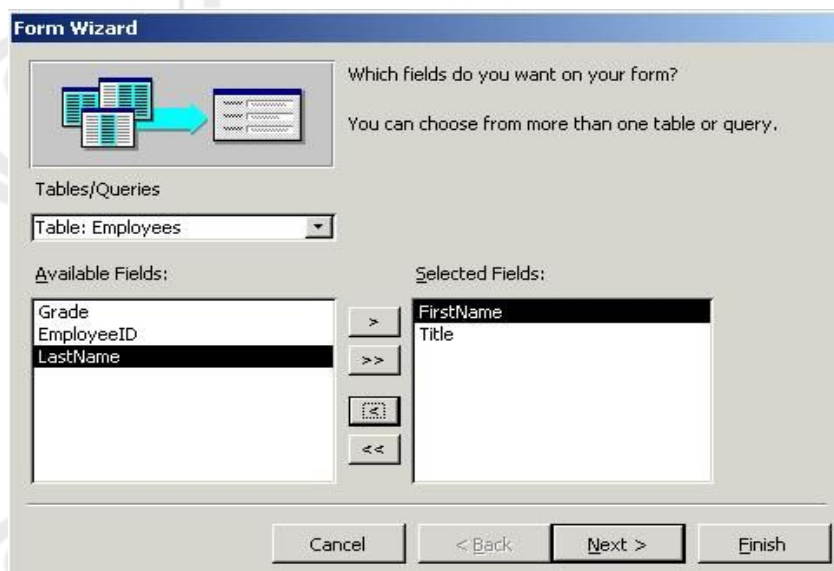


Figure 9: Create Employee Form by Using Wizard

Afterwards select the layout and visual style for the form from the next set of options and click Next. On the final screen, name the form in the space provided. Select “Open the form to view or enter information” to open the form in Form View or “Modify the form’s design” to open it in Design View. Click Finish to create the form.

To Create a Form using Design View

To create a form from scratch without the wizard, follow these steps:

Select “Design View” and choose the table or query the form will be associated with the form from the drop-down menu. Select View | Toolbox from the menu bar to view the floating toolbar with additional options. The toolbar contains different controls as shown in Figure 10.

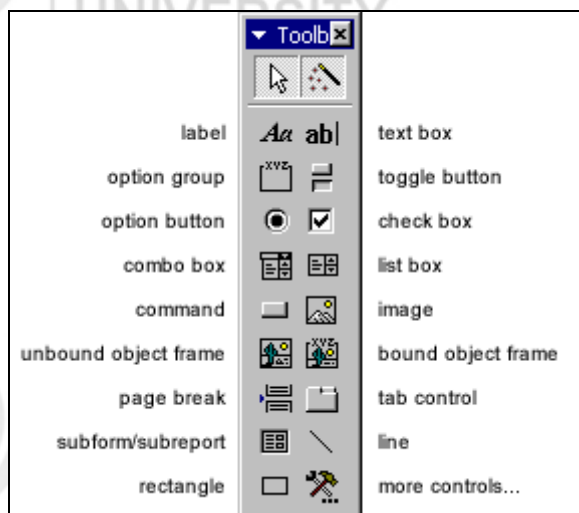


Figure 10: Different controls in Toolbar

As shown in Figure 11 you can add controls to the form by clicking and dragging the field names from the Field List floating window. Access creates a text box for the value and label for the field name when this action is accomplished. To add controls for all of the fields in the Field List, double-click the Field List window’s title bar and drag all of the highlighted fields to the form.

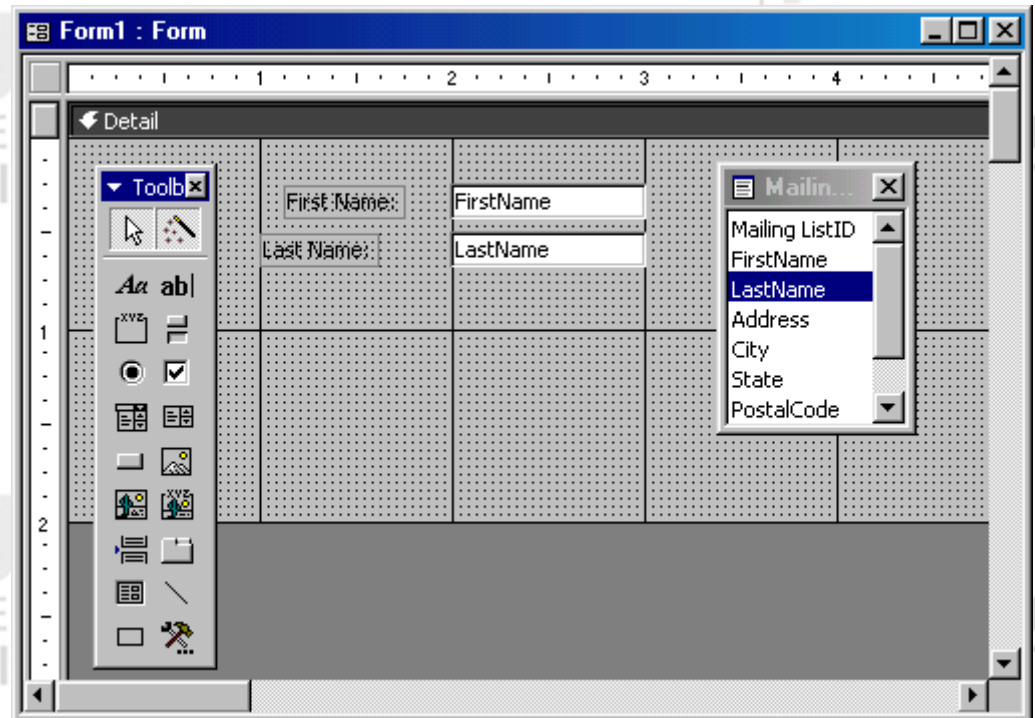


Figure 11: Adding controls to the form

A form is a customized way of viewing, entering and editing records in a database. You can specify how data is to be displayed when you design the form. Form can be created to resemble more closely the way data would be entered on paper form so that the user feels familiar with the operation.

Reports

Forms and Queries present the data on screen. Reports are used to present data on printed-paper. It provides a way to retrieve and present data as meaningful information, which might include totals and grand totals, which have to be shown across an entire set of records. Similar to Form in Reports creation also Access provides two ways for report creation. As shown in *Figure 12* you can select any way of report creation. For example in *Figure 13* you can see a report showing summary report of employee sales and category sale.

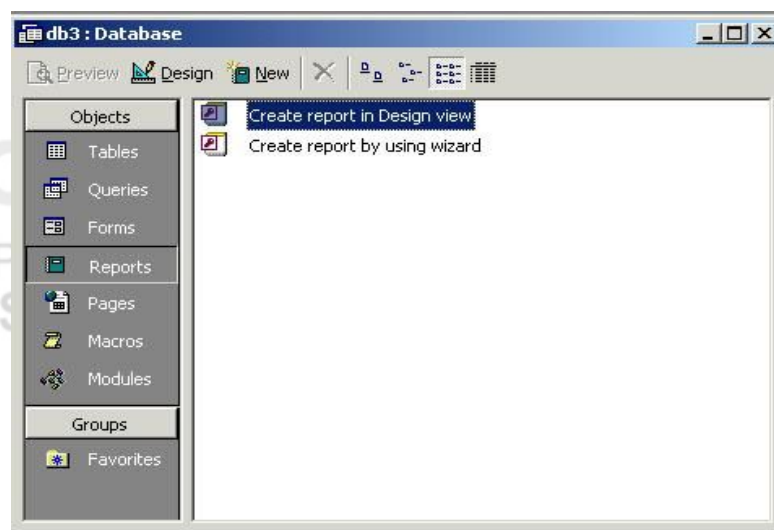


Figure 12: Creating reports

A Report

Sales Summaries		
20-Nov-04		
Employee Sales:	Employee Name:	Employee Sales:
	Pooja Sethi	47,674
	Sachin Nandwani	59,687
	Anu Solanki	93,932
Category Sales:	Category Name:	Category Sales:
	Beverages	127,189
	Condiments	56,462
	Confections	88,500

Figure 13: A sample report

1.3 DATABASE CREATION

In this section let us do some exercises relating to DBMS. The sessions are structured for your benefit.

Session 1: In this session you need to create database for an Employee management system of an ABC organisation. The details about different tables are given below. According to that you can proceed further and create tables using MS-Access.

Create the following tables with the specified constraints:

Employee

First name	-	Not NULL
Middle initials	-	
Last name	-	Not NULL
Employee-id	-	Primary Key
Date of Birth	-	
Address	-	
Gender	-	M or F
Salary	-	Range of 5000 to 25000
Date of Joining	-	
Department number	-	Refers to Department Number of Department table.

Department

Department name	-	Not NULL unique
Department number	-	Primary Key
Manager_id	-	Refers to employee-id of employee table.
Manager date of joining	-	Not NULL.

Department location

Department number	-	Refers to Department number of department table.
Department location	-	Not NULL.
Department number & Department location are combined Primary Key.		

Project

Project name	-	Not NULL.
--------------	---	-----------

Project number	-	Primary Key.
Project location	-	Not NULL.
Department number	-	Refers to department number of Department table.

Works-on

Employee-id	-	Not NULL refers to employee-id of employee table.
Project number	-	Not NULL refers to Project number of Project table.
Hours	-	Not NULL.
Employee-id & Project number are combined primary key.		

Dependent

Employee-id	-	Refer to employee table employee id field
Dependent name	-	
Gender	-	M or F
Date of Birth	-	Not NULL
Relationship	-	Not NULL

Now enter a few sets of meaningful data and answer the following queries.

1. List the department wise details of all the employees.
2. Find out all those departments that are located in more than one location.
3. Find the list of projects.
4. Find out the list of employees working on a project.
5. List the dependents of the employee whose employee id is '001'

Session 2:

This session is similar to the previous one, but in this session assume that you are developing a prototype database of the IGNOU library management system, for that you need to create the following tables:

- (a) Book Records
- (b) Book details
- (c) Member details and
- (d) Book issue details

Structure of the tables are given below:

Table Name Attribute Name

Book Records Accession Number
ISBN Number

Books ISBN Number
Author
Publisher
Price

Members Member Id
Member Name
Maximum Number of books that can be issued
Maximum Number of days for which book can be issued

Book Issue Member Id
Accession Number

You must create constraints, including referential integrity constraints, as appropriate. Please note accession number is unique for each book. A book, which has no return date, can be considered as issued book. Enter suitable data into the tables. Now answer the following:

1. Insert data in all the three tables (use insert).
2. Insert appropriate description associated with each table and the column (use comment).
3. Display the structure of the tables.
4. Display the comments that you have inserted.
5. Using SELECT statement, write the queries for performing the following function:
 - (a) Get the list of all books (No need to find number of copies)
 - (b) Get the list of all members
 - (c) Get the Accession number of the books which are available in the library
 - (d) On return of a book by a member calculate the fine on that book.
 - (e) List of books issued on 01-Jan-2005
 - (f) Get the list of all books having price greater than Rs. 500/-
 - (g) Get the list of members who did not have any book issued at any time.
 - (h) Get the list of members who have not returned the book.
 - (i) Display member ID and the list of books that have been issued to him/her from time to time.
 - (j) Find the number of copies of each book (A book accession number would be different but ISBN number would be the same).
 - (k) Find the number of copies available of a book of given ISBN number.
 - (l) Get the member ID and name of the members to whom no more books can be issued, because they have already got as many books issued as the number for which they are entitled.

1.4 USE OF DBMS TOOLS/ CLIENT-SERVER MODE

Session 3:

This session is based on Session 2 where you have created a library management system. In this session you have different **query specification**. You must create appropriate forms, reports, graphs, views and data filtering, use of multilevel report, etc. to answer these queries.

1. Get the list of ISBN-Number, Book name, available copies of the books of which available copies are greater than zero.
2. Get the list of ISBN-Number, Book name, Total copies, available copies of the book of which available copies are greater than zero. List should be displayed in alphabetical order of book name.
3. Get the list of ISBN number, Book name, Author, total copies, cost (cost is price \times total copies). List should be displayed in descending order of cost.
4. Get the list of books issued to each member.
5. Write query to know the maximum and average price of the books.
6. Get the list of all existing members and the number of days for which a member is allowed to keep the book. Also find out the members who have got the maximum number of books issued.
7. Get the list of *member codes* of those members who have more than two books issued.

8. Find the details of the books presently issued to a member.
9. Create the history of issue of a book having a typical accession number.
10. To set the width of the book name as 35.

Session 4:

Create the following table and perform the necessary tasks defined below one by one. You must use the query tools/ SQL/ Reports/ Forms/ Graphs/Views/ using client/server wherever needed.

1. Create the following table named **customer**

Column name	type	size
Customer id	Character	10
Name	Character	25
Area	Character	3
Phone	Numeric	7

Insert the appropriate data into table.

- a. Update Phone numbers of all customers to have a prefix as your city STD Code
 - b. Print the entire customer table
 - c. List the names of those customers who have 'e' as second letter in their names.
 - d. Find out the Customer belonging to area 'abc'
 - e. Delete record where area is NULL.
 - f. Display all records in increasing order of name.
 - g. Create a table temp from customer having customer-id, name, and area fields only
 - h. Display area and number of records within each area (use GROUP by clause)
 - i. Display all those records from customer table where name starts with 'a' or area is "abc."
 - j. Display all records of those where name starts with 'a' and phone exchange is 55.
2. Answer the following queries using Library system as created earlier. You must create a view to know member name and name of the book issued to them, use any inbuilt function and operators like IN, ANY, ALL, EXISTS
 - a. List the records of members who have not been issued any book using EXISTS operator.
 - b. List the members who have got issued at least one book (use IN / ANY operator).
 - c. List the books which have maximum Price using ALL operator.
 - d. Display Book Name, Member Name, Issue date of Book. Create a view of this query of the currently issued books.
 3. Create a table of Employee (emp-number, name, dept, salary) and Department (dept number, dept name). Insert some records in the tables through appropriate forms having integrity checks. Add some records in employee table where department value is not present in department table. Now answer the following query:
 - a. Display all records from employee table where department is not found in department table.
 - b. Display records from employee table in a report format with proper headings. This report must also contain those records where department number does not match with any value of department table.

- c. Display those employee records who have salary less than the salary of person whose empcode= 'A100'.
- d. Create another table : Sales_data (Region-code, City, Salesperson-code, Sales-qty).
- e. Display records where salesperson has achieved sales more than average sales of all sales persons of all the regions.

Session 5:

For the following queries use Library System as created by you in earlier sessions. You must use the query tools/ SQL/ Reports/ Forms/ Graphs/Views/ using client/server wherever needed.

1. Get the list of books presently issued to the members, along with the names of the book as well as names of the members.
2. Get the list of the members who
 - (a) are entitled for more books than that the entitlement of member name "abc".
 - (b) are issued the books for more days than the number of days for "abc".
3. Find out the history of issuing of a list of books that has been identified during inspection as damaged books. (Create the necessary tables if needed).
4. Create the tables Item master and Transaction having following format:

<u>Item Master:</u>	<u>Transaction:</u>
Item-code	item-code
Item-name	Quantity
Price	Date of transaction

Set the foreign key constraints in the tables and insert at least 5 records having meaningful data. Now answer the following queries.

- a. Display Item-code, Name, Quantity, Date of transaction, where Sales amount = Quantity *Price using a report.
- b. Display all transactions of item 'X' using a report.
- c. Display all the items whose price is more than the price of item 'X'.
- d. Saving the previous query into a temporary file.
- e. Store the database in a new file.
- f. Create all the forms for data entry and create at least 5 meaningful reports.

1.5 FORMS AND PROCEDURES

This section covers design and implementation of different kinds of forms to create user interactivity. Also, you can design different procedures/triggers to perform different operations on databases.

Session 6:

1. Create the following tables:

Order party: (Order number, Order date, customer code)
 Order: Order number, Item code, Quantity

The key to the second table is order-number + item-code
 Create a **form** for data entry to both the tables.

2. Create a form for storing Bio-data of students. Create the supporting tables to store the data.

3. Design a suitable form for storing basic information and salary details of employees of an organisation. Design and implement the necessary tables.

Session 7:

1. Write a procedure/trigger on department code so such that the validity of the code is checked and the name of department is automatically displayed on selection of department code. Assume, design and create the necessary tables and constraints.
2. Write a procedure/trigger on a numeric field named value1 to check if the entered value is 1 (Married) or 2 (Unmarried). In case, the entered value is 1 (Married) then the control should pass to a field named 'spouse name' or else it goes to a field named: Father's Name.
3. Employee code must begin with 'P' (Permanent) or 'T' (Temporary) and its second character must be a digit. Write procedure/trigger to check if the entered value is correct.
4. Write a procedure/trigger to generate Order Number automatically in any of the order tables created in Session 6.

Session 8:

1. Design a form that shows the status of books pending on a member on entering the member-id.
2. Design a **form** that modifies the records of an Item Table having the fields: Item Code, Item Name, Quantity, Price, Re-order Level.
 - (a) Enter the Item Code and get all the details from the tables
 - (b) Check if negative values are entered in the field.
3. Design the **form** to display the leave information of each employee following. The validations must be made for the fields:
 - Leave information of every employee must be display grouped by month
 - Display total of all leave taken.

Let us now perform all the operations you have practiced till now. You must use the query tools/ SQL/ Reports/ Forms/ Graphs/Views/ procedures/ using client/server wherever needed.

Session 9:

1. Add one more table employee with fields employee-number, employee-name, Basic pay, Department in the Library management system.
2. Add a new column Date of Joining in the table.
3. Modify the length of field employee name.
4. Delete the column basic from basic pay.
5. Find the details of members who were issued a book prior to Feb 1st 2005.
6. In previous query 5, list the details of the particular members.
7. In previous query 5, list the details of only two such members.
8. List the details of the persons who have been issued at least one book.
9. List the names of three persons who have not been issued any book.
10. List of members, who are entitled for 5 books or are issued the books for 15 days.
11. List the names of members in fixed length of 30 characters followed by their codes in parenthesis and with first character of the name in capital.
12. Find the list of the members who have been issued the books having the same ISBN number.

13. Display book issue/return data of various books in the following form

Book Accession number. Book Title Issued on Returned on

Session 10:

1. Create the following tables for a video library after normalizing these tables:

Customer

Customer_id	Primary Key	Not NULL
Name		Not NULL
Area		
Phone_number		

Movie

Movie_no	Primary Key	Not NULL
Title	<film title>	Not NULL
Type	Action or Thriller or Romance or Comedy or Suspense or Horror etc.	
Actors	Not NULL	
Rent-Price	Not NULL	
Rent applicable	data part of primary key	

Issues

Issue_no	Primary Key	Not NULL
Movie_no	Refers to Movie_no of movie table	
Customer_id	Refers to Customer_id of Customer table	
Issue_date	not greater than current date.	
Return_date	not greater than current date.	

Write down SQL statements to perform the following Queries:

- List the names of all the customers.
- Print the entire customer table.
- List the name and area of all the customers.
- List the various movie types available.
- List the names of all customers having 'i' in any case as the second letter in their names.
- List the names of all customers that begin with 's' or 'j'.
- Print the list of employees whose phone numbers have area code as 011.
- Print the information of customers who have been issued movies in the month of February.
- List the movies that have been issued to the customers with customer-id between '9000' and '9999'.
- List the names of movies whose Rent - price is greater than Rs. 100/-.
- Increase the Rent-price of each movie by 10%. Modify rent applicable data suitably.
- List the movies in sorted order of their title, and types of all the movies except Drama.
- Find the recovery made from each movie.
- Calculate the total revenue of all movies.
- Determine the maximum and minimum movie prices and Rename the title as Maximum Price.
- List the Movies which are issued to customers for more than a week.
- Print the type, average price, total number of prints, for each type of movie.
- Find out the movies issued to customer 'X'.

19. Find out the names of the movies that have been issued to the maximum number of customers.
20. Display the month in which customers take the maximum number of movies.
21. Display the history sheet of each movie.
22. List the customers who have not been issued any movie in the last 6 months.

1.6 SUMMARY

This section has provided you with problems with respect of creation of database and integrity using constraints and using an interface and also using SQL commands. Some of the exercises provided include creation of forms and reports, creation of SQL queries and an overview of various databases related concepts. We hope by now you must be familiar with at least one database application and would be able to migrate to other DBMSs.

1.7 FURTHER READINGS

Reference Books

- MS-Access user guide.
- Microsoft Access 2000 Bible by Cary Prague and Michael Irwin, IDG Books.
- Access 2003 Bible by Cary N. Prague, Michael R. Irwin, Jennifer Reardon; John Wiley & Sons publication.

Web references and tutorials

<http://mis.bus.sfu.ca/tutorials/MSAccess/tutorials.html>
<http://www.aspfree.com/c/b/Microsoft-Access/>
<http://netforbeginners.about.com/od/msaccess/>
<http://www.vbtutor.net/vbtutor.html>
<http://www.w3schools.com/sql/default.asp>
<http://sirius.cs.ucdavis.edu/teaching/sqltutorial/>

SECTION 1 C + + PROGRAMMING LAB

Structure	Page Nos.
1.0 Introduction	5
1.1 Objectives	5
1.2 General Guidelines	6
1.3 Some C + + Compilers	6
1.4 Dev C + + IDE	7
1.5 Using Dev C + + IDE	10
1.6 Running Example Programs Using Dev C + +	15
1.7 Session Wise Problems	18
1.8 Summary	22
1.9 Further Reading	22

1.0 INTRODUCTION

Our attempt in this section is to make you familiar with object oriented programming (OOP) using C + + . We will discuss about how to write and run C + + program. It is essential for you to practice as much as possible for getting more and more understanding in the programming and problem solving. So far you have learnt basics of problem solving and data handling concepts in courses MCS-011 & MCS-023 and their respective lab work. By now you must have experienced that by experimenting with the tool/package/programming language, you gain valuable experience and your efficiency has increased.

By now, you have obtained the practical skills of c programming languages. In C + + , almost the same kind of conditional and logical statements are used as you have used in c programming. To begin with you may consider the C + + programming as a extension of c programming. To become a good C + + **programmer**, it is essential to take one **step at a time**, while programming/ problem solving. Hence, during the problem solving/ programming process, you need to have good understanding of following:

- the tools you are using for program writing and running,
- basic language feature like data types, variables, loops and functions etc.,
- understanding of problem and deciding how to go from an idea to code, and
- a clear, understanding of OOP features need to be used in specific program.

In this lab course, we have used Dev-C + + , for writing and running the programs. The Dev-C + + , is developed by Bloodshed Software. It is a full featured graphical IDE (Integrated Development Environment), which is able to create Windows or console-based C/C + + programs using the MinGW compiler system. We have also added practice problems in this section, which you can attempt and *enjoy C + + programming*.

1.1 OBJECTIVES

By the end of the practical sessions of this section, you will be able to:

- use basic constructs of C + + in your program;

- define and use objects in C++ programs;
- define and use constructors and destructor in C++ programs;
- use concept of inheritance and polymorphism in C++ programs;
- write programs for I/O handling in C++;
- create your own C++ templates; and
- handle exceptions in C++ programs.

1.2 GENERAL GUIDELINES

- You should come prepared for your lab session to properly utilize the maximum time of Lab session.
- You should attempt all lab exercises/assignments given in the list (session wise).
- You may seek assistance in doing the lab exercises from the available lab instructor.
- For each program, you should add general comments such as your name date of programming etc.
- There should be proper comments for description of the problem, requirement of class, function etc. Proper comments are to be provided as and when necessary in the programming.
- Your program should be interactive and properly documented with real Input/Output data.
- Proper management of file of Lab record is necessary. Completed lab assignments should be submitted in the form of a Lab Record in which you have to write the algorithm, program code along with comments and output for various inputs given.
- There are total 10 lab sessions (of 3 hours each) for this lab course.

1.3 SOME C++ COMPILERS

For writing C++ program, you are free to use any C++ compiler and editor available at your centre/home. If you get opportunity you may experiment with more than one C++ compilers. In your C programming lab, you might have used Borland C/C++ compiler. The use of Borland C/C++ compiler is explained to you in BCSL-021 course. List of some compilers which are available for free download is given below. You are advised to check their conditions/licenses before using them.

- Apple C++ .
- Bloodshed Dev-C++ .
- Borland C++
- Clang C++ ..
- Cygwin (GNU C++)
- Digital Mars C++
- DJ Delorie's C++ development system for DOS/Windows (GNU C++)
- IBM C++ for IBM power, System Z, Bluegene, and Cell.
- Intel C++ for non-commercial development Sun Studio.

Here you are re-introduced about working with Borland C/C++ compiler. For more details you may refer to your BCSL-021 course.

Using Borland C/C++ Compiler

You may use the Borland C/C++ compiler during the lab sessions under MS-DOS connecting through Windows. A C++ program written in Borland C++ is saved with .cpp extension.

To start Borland C/C++

Click the **Start** button in the bottom left hand corner of the screen. The **Start** menu pops up. Select **Programs** from the **Start** menu. Select **Borland C/C++** from the **Programs** menu. Select **Borland C/C++** from the **Borland C++** menu. In summary the steps to launching **Borland C/C++** are:

Start--->Programs--->Borland C++ --->Borland C++

You should now proceed to the main window for the C/C++ development environment.

Editing and Running C++ Program

You can create a program by entering text that corresponds to C++ statements into a file. The compilation and running process is same as explained in BCSL-021 course. It is no note that Borland compilers for C/C++ runs on DOS prompt (Non-window). Borland compiler for C/C++ is not having good interface compared to Dev C++ IDE. The Dev C++ IDE runs on windows and is easy to use as it provides very good GUI.

1.4 DEV C++ IDE

Dev-C++, is a *fully featured easy to use* graphical IDE (Integrated Development Environment), which is able to create Windows or console-based C/C++ programs using the MinGW compiler system. MinGW (Minimalist GNU for Windows) uses GCC (the GNU g++ compiler collection), which is essentially the same compiler system that is in Cygwin (the unix environment program for Windows) and most versions of Linux. The GNU General Public License (GPL) is a free, copyleft license for software.

The Dev-C++ is available as free software (under GPL). Link to Bloodshed Dev-C++ sites is : <http://www.bloodshed.net/devcpp.html>.

You may download Dev-C++ and should install it in the default directory of C:\Dev-Cpp. Hence, later on installation of add-ons or upgrades can be easily done.

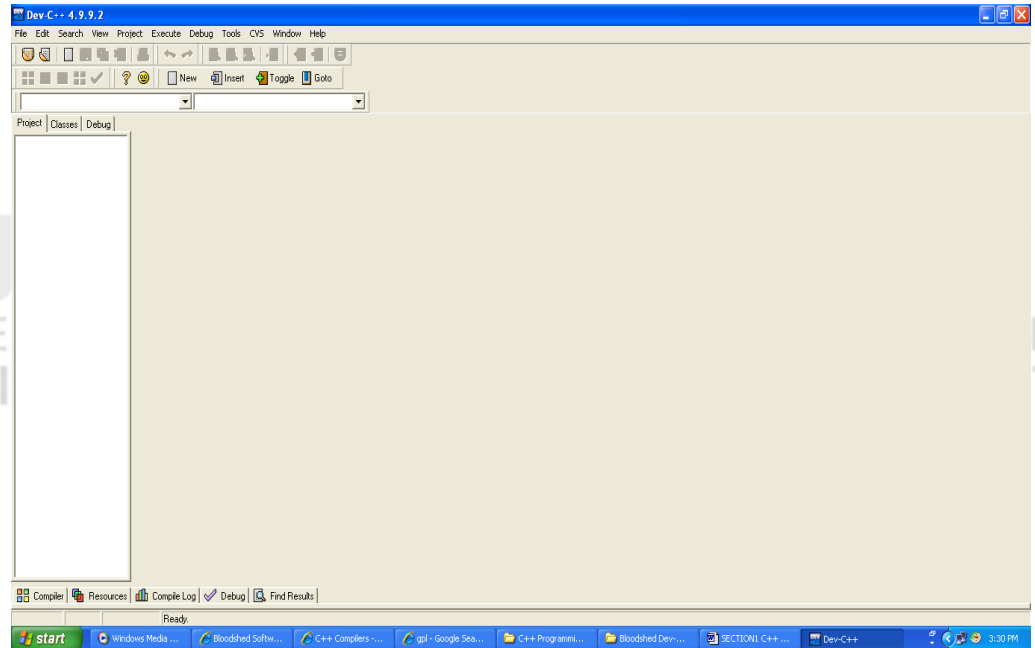
Dev C++ Installation:

To install Dev-C++ onto your computer, first download the free installation program, copying it to a folder on your computer.

1. Then install Dev-C++ onto your computer by following the steps:

- Run the setup program.
- BE SURE to use the default directory for installation (C:\dev-cpp).

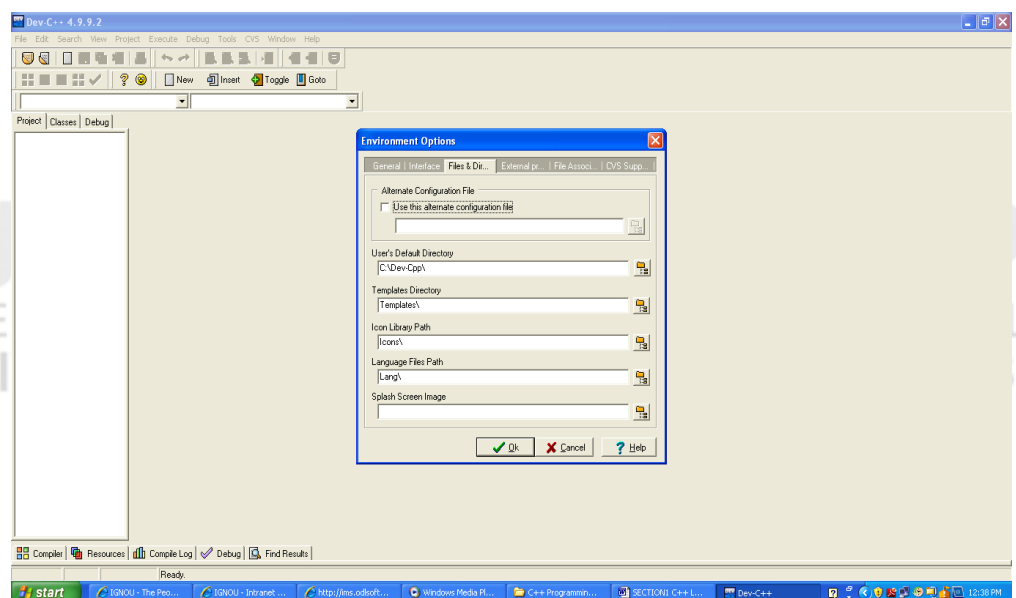
- If you insist on using a different directory, BE SURE to use a directory path that DOES NOT CONTAIN SPACES in the pathname (this means you can't use a subdirectory of C:\Program Files). Otherwise, your good C++ programs will not compile or run and you will wonder why. Spaces in pathnames to your project and source code files don't seem to matter.
- Run Dev-C++, you will get following window:



- When first time you run Dev-C++, it will ask you for your language.
- Chose English as language.

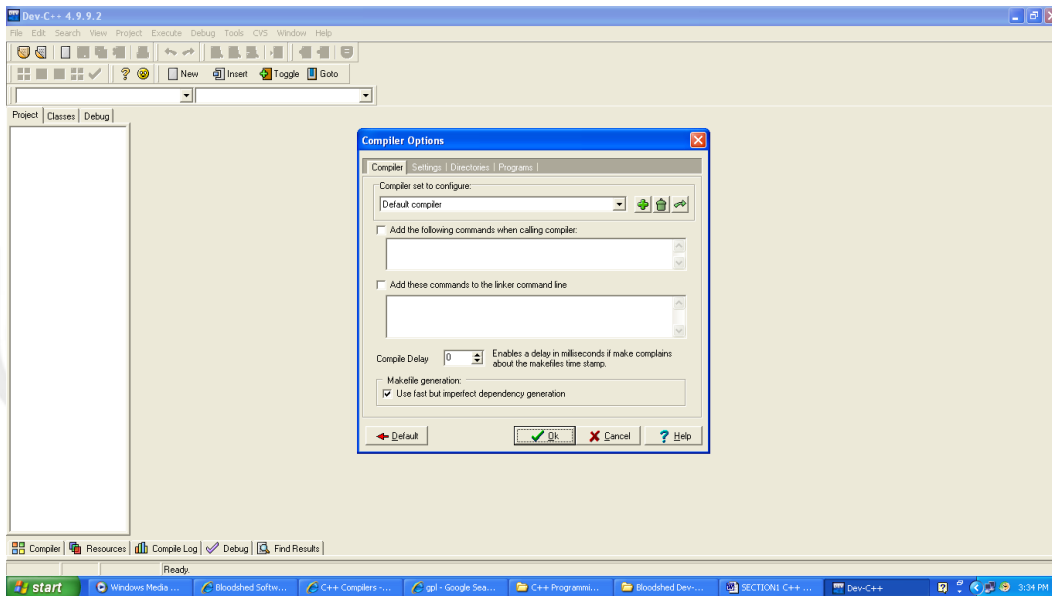
2. Now you may set the default user directory for saving/keeping your C++ programs.

- Under Tools | Environment Options, click the Files & Directories tab.

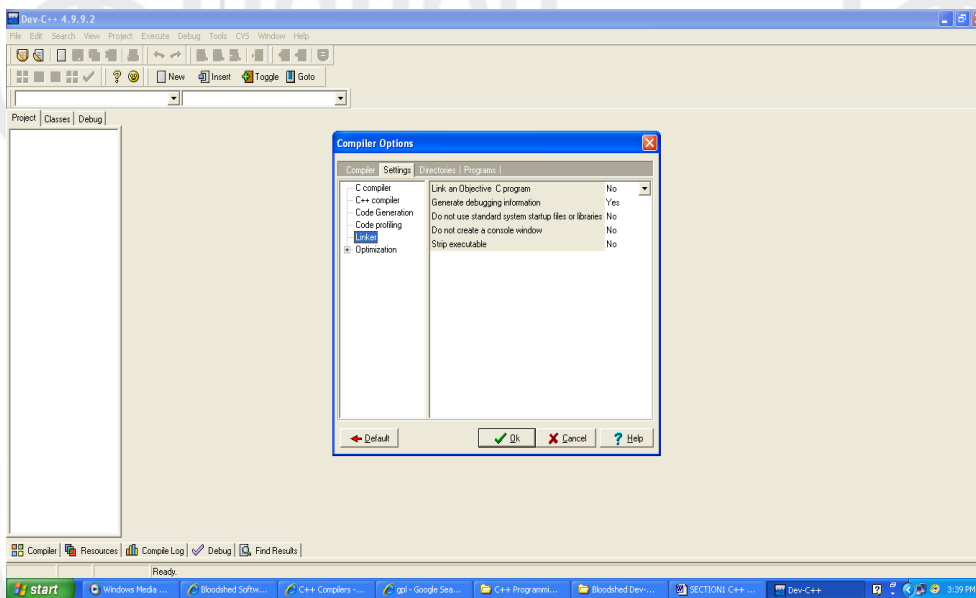


- In the "User's Default Directory" field, either type the path of your choice
- Leave all other defaults "as is".

- Click the Ok button.
3. Before you start using Dev-C++ , you need to modify one of the default settings to enable you to use debugger with your programs:
- Go to the "Tools" menu and select "Compiler Options".



- In the "Settings" tab, click on "Linker" in the left panel, and change "Generate debugging information" to "Yes":



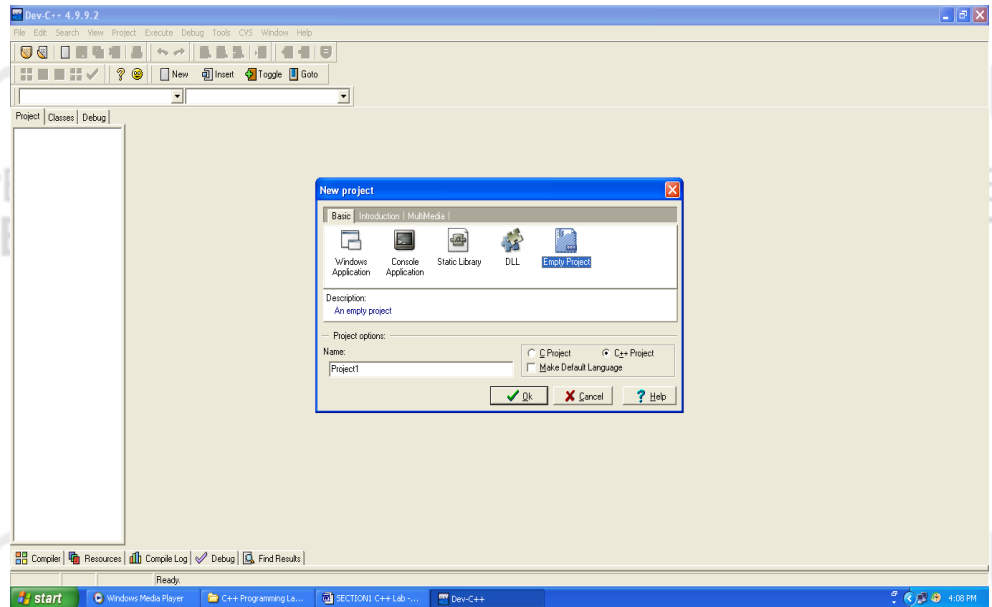
- Click "OK".

1.5 USING DEV C++ IDE

Create a new project/program in Dev-C++ :

Now you can use Dev-C++ for creating a “project/program”. A C++ project is like a container that is used to store all the elements that are required to compile a program.

- Go to the "File" menu and select "New", "Project".
- Choose "Empty Project" and make sure "C++ project" is selected.



Here you have to give your project a name. You should give your project a valid and meaningful filename. The name of your project will also be the name of your final executable.

A project may have more than one file in it. To add empty source files, follow one of two ways:

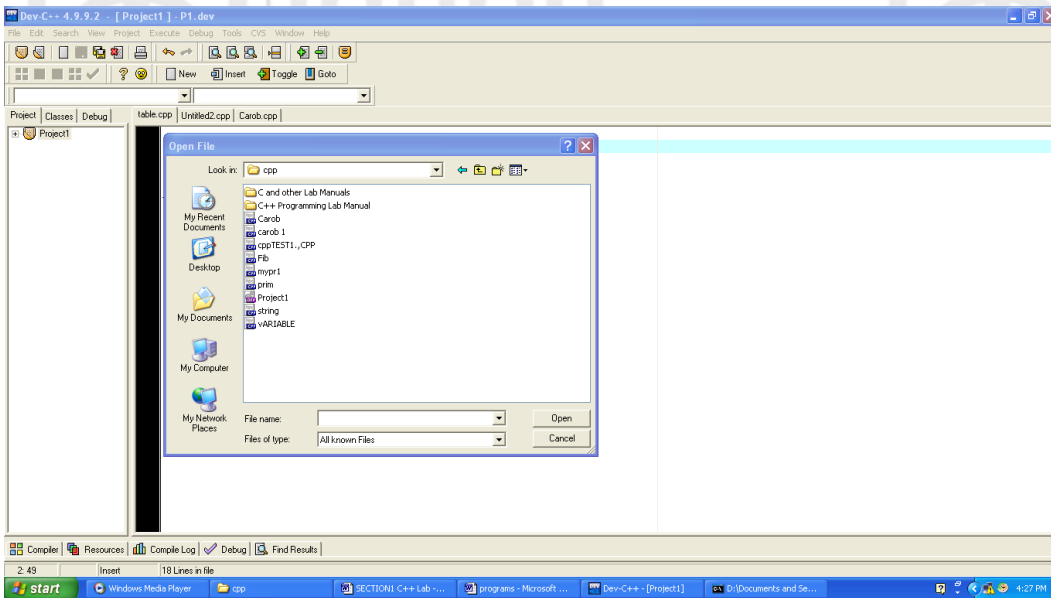
- Go to the "File" menu and select "New Source File" (or just press CTRL+N)
- Go to the "Project" menu and select "New File".

You should save your file before compilation. Note that Dev-C++ will not ask for a filename for any new source file until you attempt to do any of the following activity:

1. Compile
2. Save the project
3. Save the source file
4. Exit Dev-C++

If you are having a pre-existing source files and want to add it to your project. You can do it with one of two following ways:

- Go to the "Project" menu and select "Add to Project"
- Right-click on the project name in the left-hand panel and select "Add to Project".

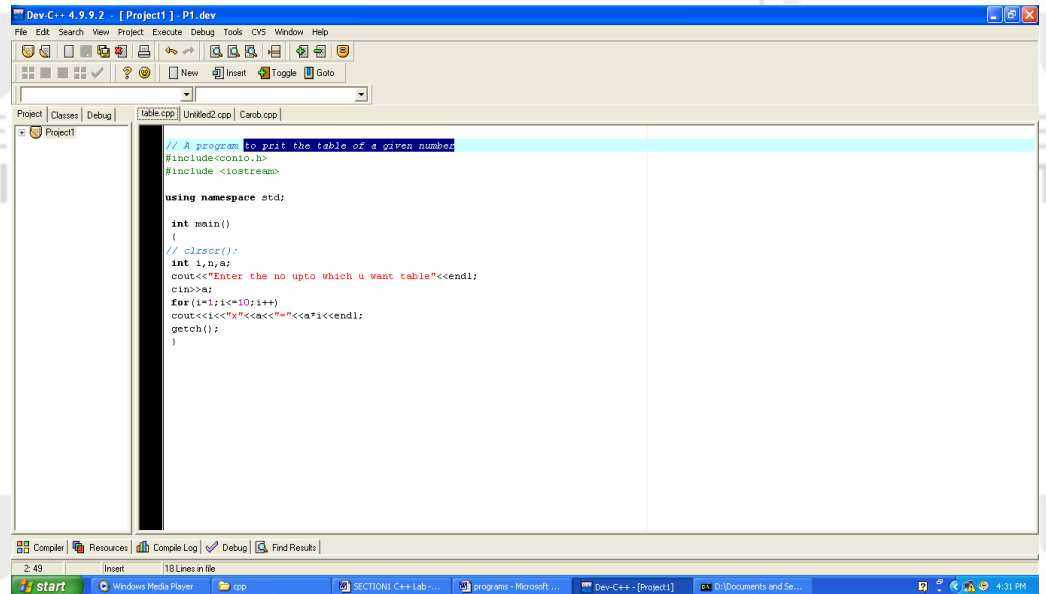


Now let us write our first program using Dev C++ . This program is to display message ***“Welcome to C++ programming”***.

```
//Program to Print Hello message
#include <iostream>
using namespace std;
int main(int n, char* Args[])
{
    char name[20];
    cout << "Please enter your name:";
    cin>>name;
    cout << "Hello Mr./Miss. "<<name;
    cout<<" Welcome to C++ programming"<<endl;
    system("PAUSE");
    return 0;
}
```

The above program include the statement, **"using namespace std;"**. This line tells the compiler to use a group of functions that are part of the standard library (std). By including this line at the top of a file, you allow the program to use functions such as cout.

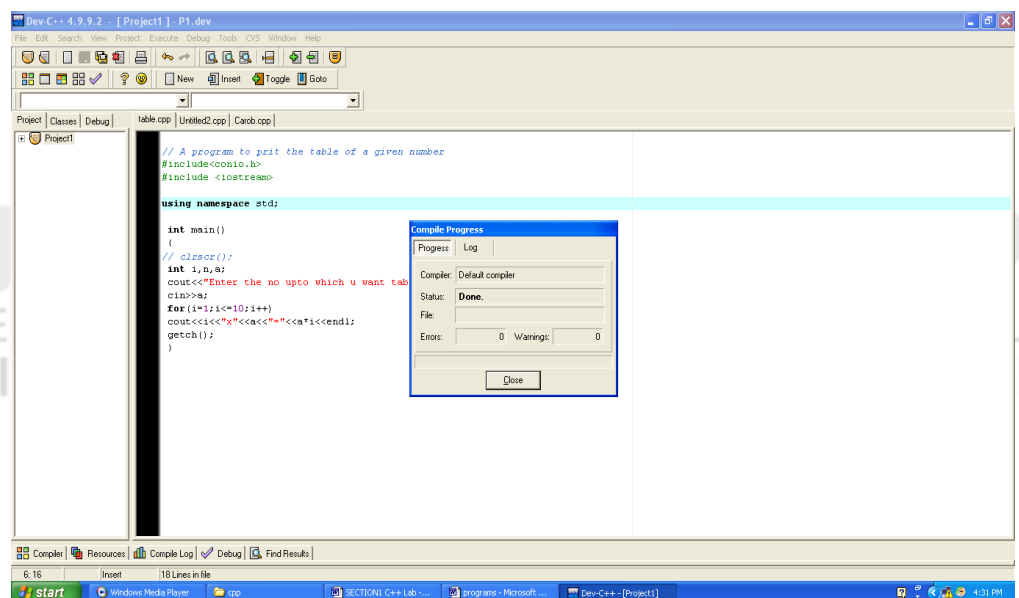
Let us try to execute above programe in Dev C++ . Type this program in Dev C++ IDE. Once you complete the writing, save the file with .cpp extention. Now you program is ready for compilation.



Compile program in Dev C++ :

Once your program coding is over, you are ready to compile.

- Go to the "Execute" menu and select "Compile" (or just press CTRL+F9).
- There is a chance that you will get some kind of compiler or linker error the first time you attempt to compile a project.
- Syntax errors will be displayed in the "Compiler" tab at the bottom of the screen.
- You can double-click on any error to take you to the place in the source code where it occurred.
- The "Linker" tab will flash if there are any linker errors. Linker errors are generally the result of syntax errors not allowing one of the files to compile.
- Once your project successfully compiles, the "Compile Progress" dialog box will have a status of "Done". At this point, you may click "Close".

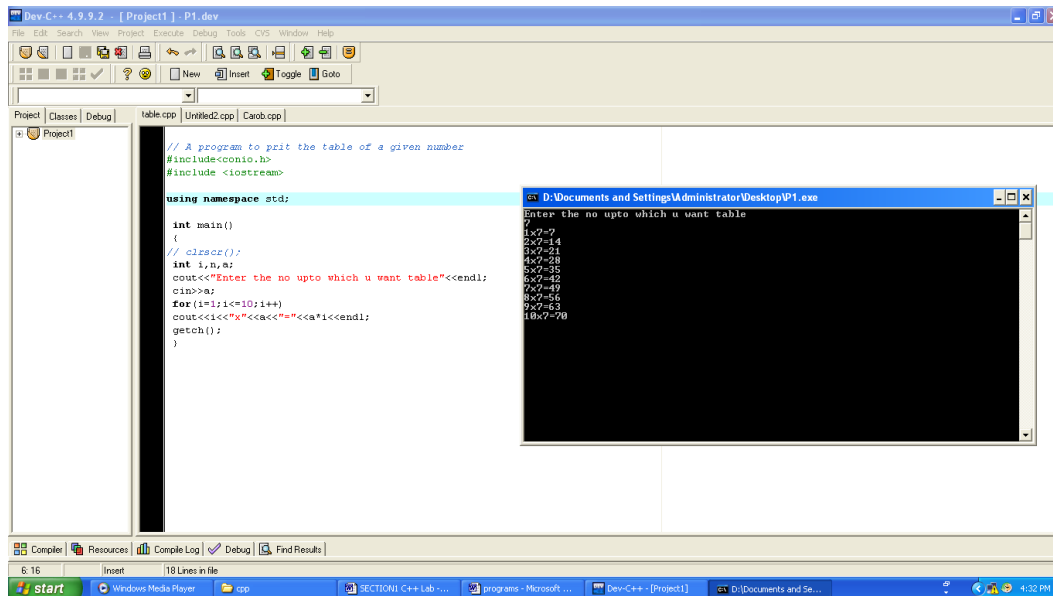


Executing Program in Dev C++ :

You can now run your program.

- Go to the "Execute" menu, choose "Run".

Note: If you have to pass command-line parameters to your program, go to the "Execute" menu, choose "Parameters" and type in any parameters you wish to pass.



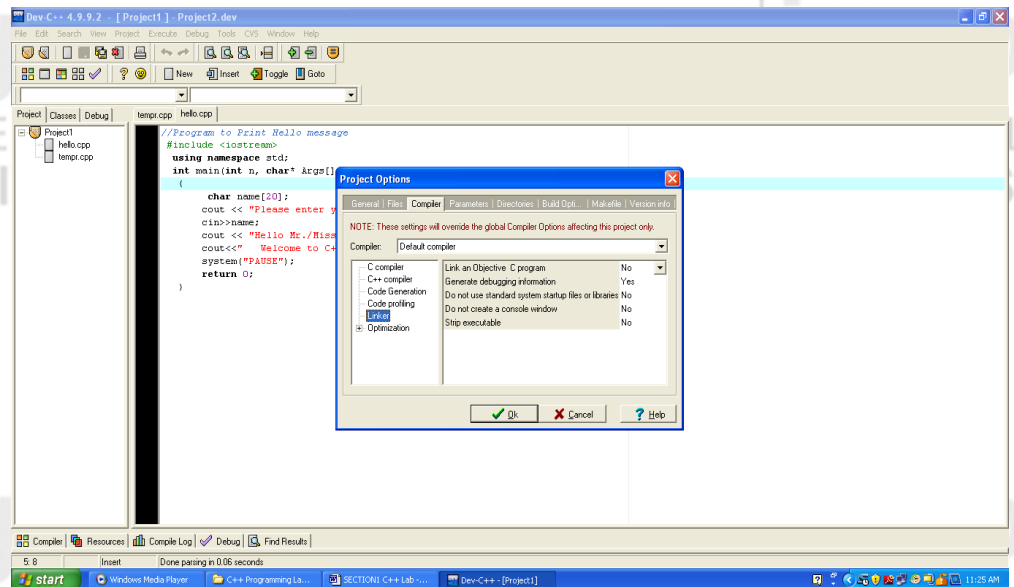
Removing Disappearing windows Problem:

If you execute your program, you may notice something peculiar; a console window will pop up, flash some text and disappear. The problem is that, if programs directly executed, console program windows close after the program exits. Not that in the above program system ("PAUSE"); command is used. If this command is not used, guess what will happen? The program would print the name entered to the screen and then immediately close without giving you the chance to read what was printed. But you need to see the result/output of the program. Here is the use of system ("PAUSE"); , it give you a chance to read what was actually printed. You may try the above program after removing/commenting, system ("PAUSE"); and can observe the advantage of this command.

Program Debug in Dev C++ .

For using debugging facility in Dev C++ , you have to first, make sure that you are using a project. Then follow the following steps:

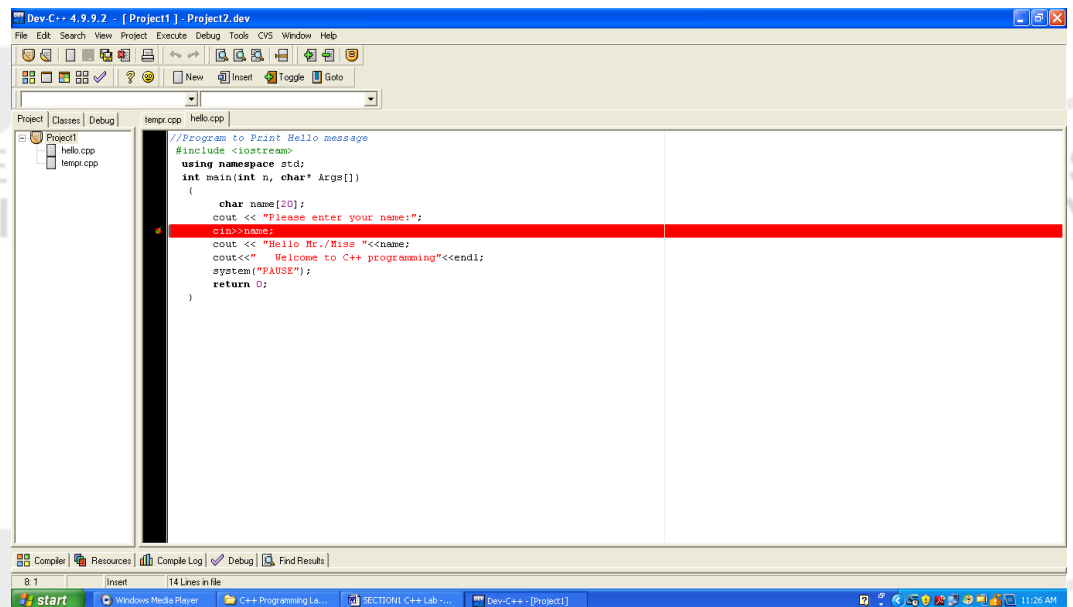
Goto Project Option> Compiler>Linker and set General debugging information to "yes".



After that, do a full rebuild (Ctrl-F11), then set breakpoint(s) where you want the debugger to stop (otherwise it will just run the program).

To set a breakpoint on a line, click on the gutter (the gray band on the left), or press Ctrl-F5.

Dev-C++'s basic debugger functions are controlled via the "Debug" tab at the bottom of the screen. More advanced functions are available in the "Debug" menu of Dev-C++.



For using Dev-C++ debugger functions you may use the following instructions:

- Click the "Run to cursor" icon to run your program and pause at the current source code cursor location.
- Click "Next Step" to step through the code.
- You can launch the debugger, by pressing F8 or clicking the debug button.
- Now you may start debugging. If execution of program goes well, the program will start executing step by step, and then stop at the first breakpoint.

- Then you may go through the code, entering function calls, by pressing Shift-F7 or the "step into" button, or stepping over the function calls, by pressing F7 or the "next step" button.
- You can press Ctrl-F7 or the "continue" button to continue execution till the next breakpoint.
- At any time, you can add or remove breakpoints.
- When the program stopped at a breakpoint and you are walking through the code, you can display the values of various variables in your program by putting your mouse over them, or you can display variables and expressions by pressing F4 or the "add watch" button and typing the expression or variable name.

A debugger basically runs a program while keeping track of its functions, variables and instructions. It is capable of stopping your program at a given moment, which is called breakpointing. It is necessary to set a breakpoint. So that the debugger may know where to stop. If you don't set a break point, the debugger runs the whole program and stop only after finishing of the program.

1.6 RUNNING EXAMPLE PROGRAMS USING DEV C++ IDE

In this section, we will practice some programming in C++ using Dev C++. As you know, A C++ program is a collection of function, object, and type declarations. Every program must have a function `int main() { ... }` where the curly braces enclose a block, a sequence of declarations and statements ending in semicolons which are executed in order to solve the given problem.

A statement is an expression, block, or control statement which is used for decision making, looping etc. such as *if, for, while, break, return*. Some types (`std::string`), objects (`std::cout`), and functions are defined in header files, requiring the line `#include <header>` before use. Items defined in the standard headers are in the namespace `std`.

In programming, before you try to receive input, you must have a place to store that input. The input and data are stored in variables. There are several different types of variables which store different kinds of information. When you tell the compiler you are declaring a variable, you must include the data type along with the name of the variable. The name of variables should be meaningful. This will help you in tracing and debugging your program. Also if some body else read your program, proper variable naming will help them to understand your program.

Also using the right variable type is important for making your code readable and efficient. You should use appropriate type when declaring variables, as some variables require more memory than others. For example to store numeric values, a float may be used to store an "inexact" value and integer may be used to store "exact" value.

Sometimes compilation errors create problems and irritation to programmers. Usually, times compilation errors may seem like a small, but as a beginner in C++ programming, you may not be familiar with the strictness of the format of the program (the syntax), hence you may tend to run into lots of complaints generated by the compiler. You avoid many of compilation errors and add more clarity to your program by following some basic principles given below:

- Just do not start learning C++ programming by compiling the examples. Try to understand the line by line code of examples given in your BCS-031 course or from some other reference locations.
- If there is something, you do not understand about a program, simply make some changes in the program and see the effect, in two to three attempts, you will come to know what actually that program is intended to do.
- With large and complex advanced programs that interact with the file system and uses advance features of OOP, you need to be much more careful about experimental program changes.
- Avoid jumping forward while program analysis. If the compiler indicates an error, or your program is not giving expect result, try to explore why this happened then try to fix the error. If a program compiles and runs without proper result, take the time to study each line and its purpose, so that problem can be traced.
- Do not write whole/many lines of code without compiling intermediate results of the program. Some time you will find that errors are unmanageable due to very large number of lines/code of the program. You may get confused to find the actual source of error. Hence, use compiling frequently to check your work.
- Upon reaching the end of main, the closing brace, your C++ program will return the value of 0 to the operating system. This return value is important as it can be used to tell the OS whether our program succeeded or not. A return value of 0 means success and is returned automatically.
- Declare a variable before its use, if you attempt to use a variable that you have not declared, your program will not be compiled or run, and you will receive an error message with information that you have made a mistake.
- C++ is case sensitive where use uppercases or lowercase letters matters. In C++, all language keywords, all functions and all variables are case sensitive. The word Age and age are two different things to the compiler. A difference in case between your variable declaration and the use of the variable in expressions/calculations will lead to undeclared variable error, while compilation.
- Give appropriate comments in program. Give comments at as many places as possible to explain structure and functionality of the program. This will also work to your advantage in ensuring more clarity and application.

Now let us write our first program to create a class in C++.

Example1: This program I create a class named Book. This class is having following private data member:

- Title
- Author
- ISBN
- Publisher
- Price

Also the Book class is having following member functions to access/display the data members:

Display_Title() : It display the title of Book

Display_Price() : It display the price of Book

Display_Book_Info(): It display the title, price, author and publisher details of Book

```

//This program creat a Book object and print the Book details
#include <iostream>
#include <string>
using namespace std;
class Book // Book Class
{
private:
    int ISBN;
    int Price;
    string Title;
    string Author;
    string Publisher;
public:
    Book( int a, int b, string c, string d, string e) // Constructor
    {
        ISBN = a;    Price= b;    Title = c;    Author = d;    Publisher = e;
    }
    void Display_Book_Info( ) //method to display book information
    {
        cout<<"Book Title: "<<Title<<endl;
        cout<<"Book Price Rs: "<<Price<<endl;
        cout<<"Book ISBN: "<<ISBN<<endl;
        cout<<"Book Author Name: "<<Author<<endl;
        cout<<"Book Publisher Name: "<<Publisher<<endl;
    }
};
int main()
{
    Book b( 11111, 50, "C++ Programmin Lab", "M.P.Mishr", "IGNOU"); // object creation
    b.Display_Book_Info(); // method invoked by the object b
    system("PAUSE");
}

```

Example2: Try the following program code to run:

```

#include <cstdlib>
#include <iostream>
using namespace std;
int main(int argc, char *argv[])
{
    int i = 'a';
    cout<<"The value of i is:"<<i<<endl;
    system("PAUSE");
    return (0);
}

```

You will get error message while compiling, because in this code an integer type variable **i** is assigned a character value which is a type mismatch.

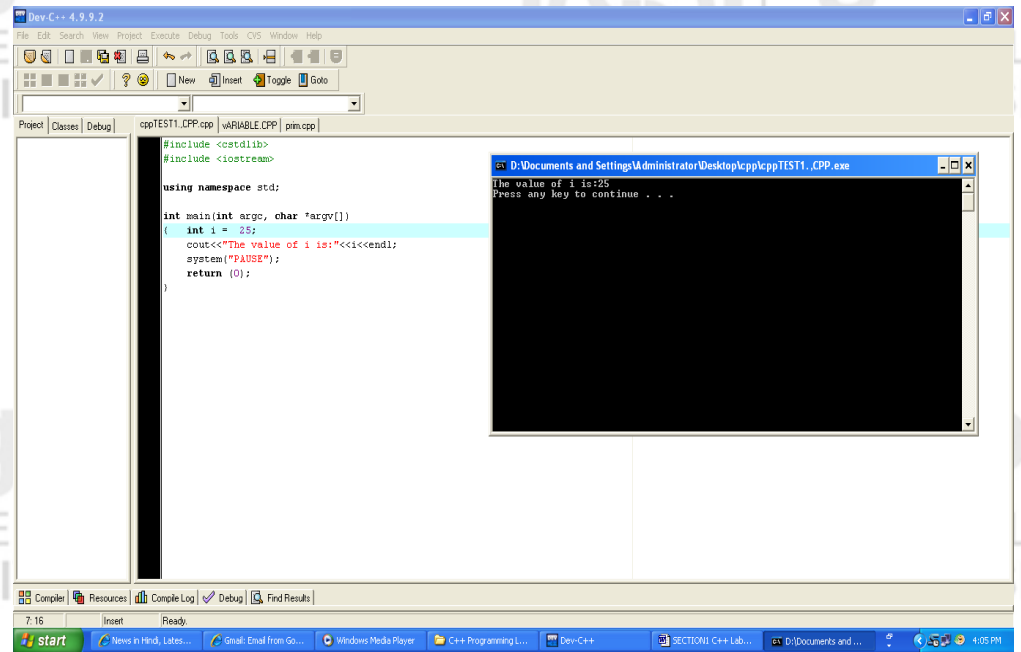
```

#include <cstdlib>
#include <iostream>
using namespace std;
int main(int argc, char *argv[])
{
    int i = a;
    cout<<"The value of i is:"<<i<<endl;
    system("PAUSE");
    return (0);
}

```

Message
 In function 'int main(int, char*)':
 7: 'a' undeclared (first use this function)
 (Each undeclared identifier is reported only once for each function it appears in.)

Now try this program by assigning an integer value to i.



1.7 SESSION WISE PROBLEMS

We have allotted **ten** practical sessions for you to exercise. You have keep all program document in the file including all the steps involve in solving these problems. You must take output of pogram and should keep in your file. You are again advised to come prepared with your design of the following problems on paper in order to take maximum advantage of the Lab session. You are advised to make assumptions wherever necessary while solving these problems.

Session 1:

Problem1. Write a program in C++ to find and print the all basic data types of C++.

Problem 2. Write a program in C++ to explain the use of for loop, while loop, swith-case, break and continue statements.

Problem 3. Write a program in C++ to find the maximu mark,average- marks and minimum marks obtained by a study in five papers as given below:

	Paper 1	Paper 2	Paper 3	Paper 4	Paper 5
Marks	50	70	65	80	56

Problem 4. Write a program in C++ that :

- prints the factorial of a given number
- prints wheatehr a number is prime or not
- generate a Fibonacci series of 50 numbers.

Session 2:

C++ Programming

- Problem 5. Write a C++ program to create Student class with appropriate constructor and destructor. Also show what happen when you try to appent to access private data members from outside of the class.
- Problem 6. Write a program in C++ to calculate the simple interest and compound interest for a given time period, given rate and for given principal amount using concept of class and member function.
- Problem 7. Write a program in C++ to create an Employee class whichg read the following information from the keyboard:

```
Employee_id
Employee_name
Designation
Employee_Department
Years of experience
and display it on screen using display_info() method.
```

Session 3:

- Problem 8. Write a program in C++ with class Rectangle with the data fields width, length, area and color .The length, width and area are of double type and color is of string type .The methods are set_length () , set_width (), set_color(), and find_area (). Create two object of Rectangle and compare their area and color. If area and color both are same for the objects then display “Matching Rectangles” otherwise display “Non matching Rectangle”.
- Problem 9. Create a class Account with two overloaded constructors using C++ . First constructor is used for initializing, name of account holder, account number and initial amount in account. Second constructor is used for initializing name of account holder, account number, addresses, type of account and current balance. Account class is having methods Deposit (), Withdraw (), and Get_Balance().Make necessary assumption for data members and return types of the methods. Create objects of Account class and use them. Also create appropriate destructors for the Account class.
- Problem 10. Write a program in C++ to create a Queue class with insert () and delete () methods. Create two objects of Queue with 10 data item in both. Also define a method to find queue lenth.

Session 4:

- Problem 11. Write a program in C++ to create a data base with the following items by using structure/class:
- ```
Name of the patient
Sex
Age
Ward_number
Bed number
Date of admission
List of Diease
```

Problem 12. Write a program in C++ to create a Player class. Inherit classes Cricket\_Player, Football\_Player and Hockey\_Player from Player class. Create appropriate constructors and destructors for these classes.

**Session 5:**

Problem 13. Write a program in C++ to generate a database with the following details about the workers working in a manufacturing company, by using structure/class:

Name  
DOB  
Blood\_Group  
Height  
Weight  
Insurance\_Policy\_No  
Office\_address  
DL\_NO

Problem 14. Write a class Book and derive classes ReferenceBook and Magazines from it. Every book has a price, title, author and ISBN number. Implement constructors, destructors and appropriate methods to manage a Book Store.

Problem 15. Write a C++ program to create Time class to implement the working of a normal watch by creating appropriate constructor, destructor and member function.

**Session 6:**

Problem 16. Write a program which reads two numbers x and y then perform the following function

- a.  $x + y$
- b.  $x - y$
- c.  $x * y$
- d.  $x / y$

Design the function  $x + y$  and  $x - y$  in base class and design the function  $x * y$  and  $x / y$  in derived class

Problem 17. Write a program in C++ which creates a Student class with methods to Display\_Grade and manage Attendance. Derive classes PG\_Students and UG\_Students from Student class.

Problem 18. Write a program by using inheritance to create the payroll of an organization by the following information:

Employee name  
Employee Designation  
Account Number  
Date of Joining  
Basic pay  
DA, HRA, CCA  
Deduction (PF, PPF etc)

**Session 7:**

Problem 19. Write a C++ program to create a class called DATE. Accept two

valid dates in the form dd/mm/yy. Implement the following operations by overloading the operators + and -. After every operation display the results by overloading the operator <<.

- i. no\_of\_days=d1-d2; where d1 and d2 are DATE objects. d1>=d2 and no\_of\_days is an integer.
- ii. d2=d1-no\_of\_days; where d1 is a DATE object and no\_of\_days is an integer.

**Problem 20.** Write a C++ program to create a class called COMPLEX and implement the following overloading functions ADD that return a COMPLEX number.

- i. ADD(a, s2)- where a is an integer (real part) and s2 is a complex number.
- ii. ADD(s1, s2)- where s1 and s2 are complex numbers.

**Problem 21.** Write a C++ program to create a class called MATRIX using a two-dimensional array of integers. Implement the following operations by overloading the operator== which checks the compatibility of two matrices to be added and subtracted. Perform the addition and subtraction by overloading the operators + and - respectively. Display the results by overloading the operator<<.

```
if(m1==m2) {
 m3=m1-m2;
 m4=m1+m2;
}
else display error
```

### Session 8:

**Problem 22.** Write a program which designs two classes and calculate the division of first class private data with second class private data by using concept of friend function.

**Problem 23.** Write a program to calculate following function by using the concept of virtual function.

$$F(x) = 7x^5 + 9x^4 + 6x^3 + 2x^2 + 4$$

### Session 9:

**Problem 24.** Write a C++ program to facilitate user to handle any chance of divide by zero exception.

**Problem 24.** Two vehicles are running on single track. If the vehicles are running in opposite direction there is a chance of collision. To avoid collisions write a C++ program using exception handling. You are free to make necessary assumptions.

**Problem 25.** Write a function template palindrome that takes a vector parameter and returns true or false according to whether the vector does not read the same forward as backward.

**Problem 26.** Create a C++ Templet class for implementation of datastructure Queue functionalities.

### Session 10:

**Problem 27 .** Implement the case study given in your course BCS-031 Block 3 Unit 4.

---

## 1.8 SUMMARY

---

This lab manual is for supporting the practice of C++ programming skill, based on BCS-031 course. This manual has detailed out how write and execute C++ programs. This lab manual started with discussion on need of a tool to write and execute C++ program. Using of the Dev C++ an IDE for C++ programming is explained with the help of explaining the steps to take in C++ program execution. Also how to debug C++ program in Dev C++ is explain in this manual. The best programming practice make a lot of impact in learning programming skill. This lab manual has hielighed how to write C++ programs to gain in terms of time saving and programming skill development. In the last section of this manual, a list of C++ programming exercised are given, which need to be solved learners by all of you for a complete understanding.

---

## 1.9 FURTHER READINGS

---

- Bjarne Stroustrup, *The C++ Programming Language*, 3rd edition,, Addison Wesley, 1997.
- John R Hubbard, *Programming with C++*, Schaum's Series, TMH publication, 3rd edition. Thinking in C++, Sunil K. Pandey, S. K. Kataria Publication, Delhi, *Object-Oriented Programming in C++*, 6th edition, Rajesh K. Shukla, Wiley India publication, 1st edition.
- [www.indiastudychannel.com](http://www.indiastudychannel.com)
- <http://cs.fit.edu/~mmahoney/cse2050/how2cpp.html>
- <http://www.cprogramming.com/tutorial/lesson1.html>
- [http://en.wikipedia.org/wiki/List\\_of\\_compilers](http://en.wikipedia.org/wiki/List_of_compilers)
- <http://www2.research.att.com/~bs/compilers.html>
- <http://ceng230.ceng.metu.edu.tr/Compilers/dev-cpp.html>