

7.1 OVERVIEW

Access Form creates the user interface to your table. Although you can use Datasheet to perform many of the same functions as forms, Forms are used as an alternative way to enter data into a database table. It provides a different way of viewing table data. Access enables us to create forms that can be used to enter, maintain, view, and print data.

The Form is constructed from a collection of individual design elements called controls or control objects. Controls are the components we see in the windows dialog boxes of the access and other windows applications like buttons, check boxes etc. We use text boxes to enter and edit data, labels to hold field names and object frame to display graphics.

A Form Wizard is provided to assist you in the construction of forms. Four types of forms can be created. These include single-column (displaying one record at a time in a vertical format), tabular (displaying multiple records in a row-and-column format), main/subform (combining the single-form and tabular formats into one form), and graph.

 Create Form by Using Wizard

To create a form using the assistance Of the wizard and 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 greater symbol button > to move one field at a time to the Selected Fields window. To move all of the fields to Select Fields, click the double greater symbol [f you make a mistake and would like to remove a field or all of the fields from the Selected  Fields window, click the left arrow < or left double arrow << buttons. After the proper fields have been selected, click the Next > button to move on to the next screen.

Microsoft Access Forms Reports 7

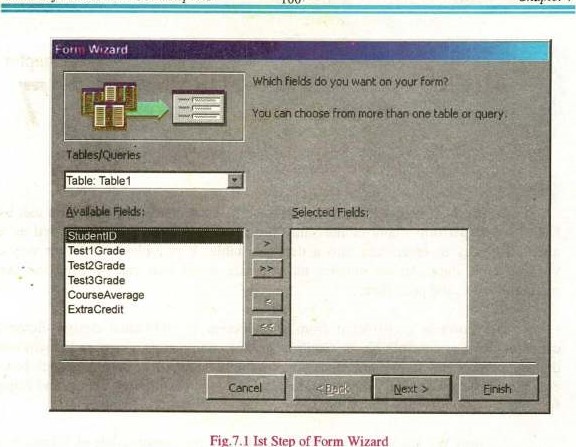


Fig.7.1

1st

Step

of

Wizard

* + - * In the second step we have the screen to select the layout of the form.

+ Columnar Form: A single record is displayed at one time with labels and form fields listed side-by-side in columns. In a columnar fom each field appears in a separate line with a label to its left; only one record is shown on each screen. The wizard fills the first column with as many fields as will fit and so forth.

+ Justified Form: A single record is displayed with labels and form fields are listed across the screen

Tabular Form: Multiple records are listed on the page at a time with fields in columns and records in rows. Tabular forms display fields in horizontal row, with field labels at the top of the form. Each new row represents a new record. Tabular forrns are best "'hen you to display just a few relatively narrow fields and you want to see sevecal records at once. To avoid spending most of your time scrolling back and forth in a tabular form, add just a few fields to the form.

+ Datasheet Form: A datasheet form initially displays data in datasheet view, much as it appears when you open a table, or run a query, or when you use the Form view toolbar button to switch to datasheet view



in any form. This type of form is often used as the basis for sub forms.

* Click the Next > button to move on to the next Step.

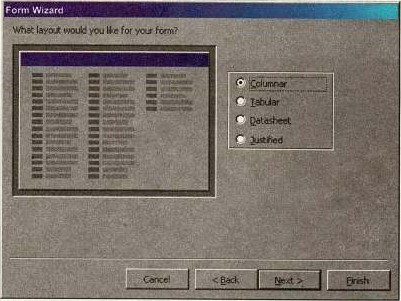


Fig.7.2 Defining Different types of Lyout

* Select a visual style for the form from the next set of options and click Next

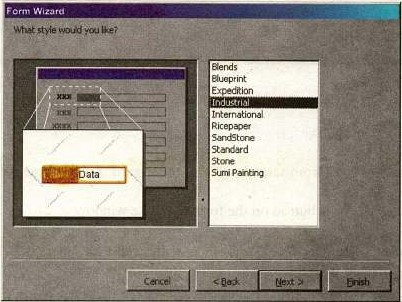
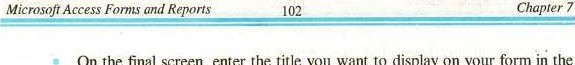


Fig.7.3 Chcxmng a Form Styles in Firm Wizard

On the final screen, enter the title you want to display on your form in the text box and than name the form in the space provided. Select "0pen 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.

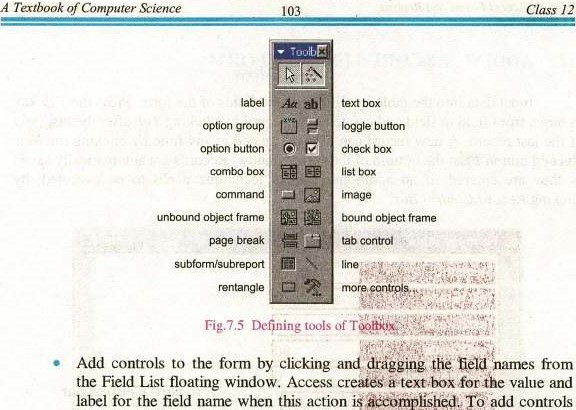
|  |  |  |  |
| --- | --- | --- | --- |
|  | | | |
| Form Wizard | | | |
|  | | | |
|  | | | |
|  | | | |
| Care |  |  |  |

Fig.7.4 Finishing the Form in Form Wizard

Create Form in Design View

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

* Click the New button on the form database window.
* Select "Design View" and choose the table or query the form will be associated with the form from the drop-down menu.
* Select View•lToolbox from the menu bar to view the floating toolbar with additional options.



action

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.

pdd

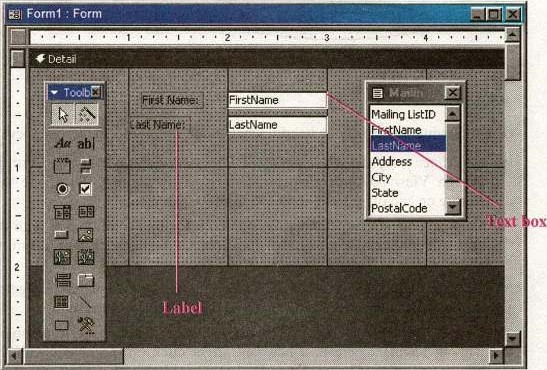


Fig.7.6 Creating a Form by Design View

|  |  |  |
| --- | --- | --- |
| Microso Access Forms and | rts | 7 |

R Chapter

7.2 ADDING RECORDS USING A FORM

Input data into the table by filling out the fields of the form. Press the Tab key to move from field to field and create a new record by clicking Tab after the last field of the last record. A new record can also be created at any time by clicking the New Record button Aat the bottom Of the form window. Records are automatically saved as they are entered so no additional manual technique needs to be executed, by clicking Record Control Bar.

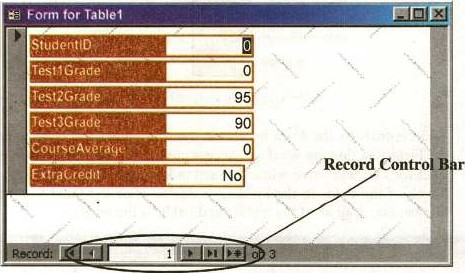


fig-7.7 Adding the record in a Form Design View

Editing Forms

The following poin%s may be helpful when modifying forms in Design View.

* Grid lines - By default, a series of lines and dots underlay the form in Design View so form elements can be easily aligned. To toggle this feature on and off select ViewlGrid from the menu bar.

Snap to Grid - Select FormatlSnap to Grid to align form objects with the grid to allow easy alignment of form objects or uncheck this feature to allow objects to float freely between the grid lines and dots.

* Resizing Objects - Form objects can be resized by clicking and dragging the handles on the edges and corners of the element with the mouse.
* Change form Object type - To easily change the type Of form Object without having to create a new one, right click on the object with the mouse and select Change To and select an available object type from the list.



* Label/object alignment - Each form object and its corresponding label are bounded and will move together when either one is moved with the mouse. However, to change the position of the object and label in relation to each other (to move the label closer to a text box, for example), click and drag the large handle at the top, left corner of the object or label.
* Tab order - Alter the tab order of the objects on the form by selecting ViewlTab Order... from the menu bar. Click the grey box before the row you would like to change in the tab order, drag it to a new location, and release the nu»use button. This can also be done by using Tab Property in the properties.

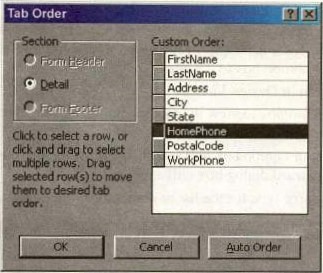


Fig.7.8 Defining the Tab Order in a Form

* Form Appearance: Change the background color of the form by clicking the Fill/Back Color button on the formatting toolbar and click one of the color swatches on the palette. Change the color of individual form objects by highlighting one and selecting a color from the Fon11Fore Color palette on the formatting toolbar. The font and size, font effect, font alignment, border around each object, the border width, and a special effect can also



be modified using the formatting toolbar:



Ljne/Border

Special

Fig.

7.9

Defining

the

Firmatting

106



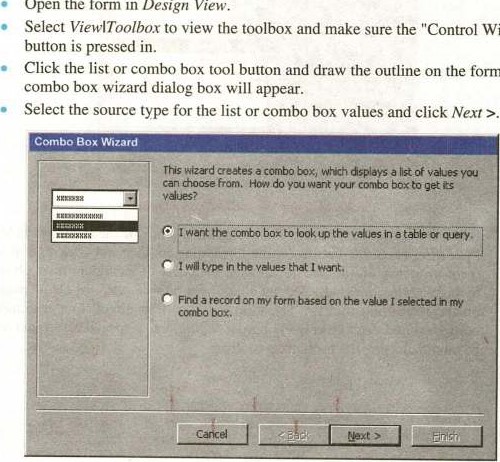
* Page Header and Footer: Headers and footers added to a form will only appear when it is printed. Select these sections by choosing ViewlPage Header/Footer on the menu bar. Page numbers can also be added to these sections by selecting InsertlPage Numbers. A date and time can be added from InserrlDate and Timem. Select ViewlPage Header/Footer again to hide these sections from view in Design View.

7.3 LIST AND COMBO BOXES

If there are small, finite number of values for a certain field on a form, using combo or list boxes may be a quicker and easier way of entering data. These two control types differ in the number of values they display. List box values are all displayed during data entering while the combo box values are not displayed until the arrow button is clicked to open it as shown in these examples:

By using a combo or list box, the name of the cities does not need to be typed for every record. Instead, it simply needs to be selected from the list. Follow these steps to add a list or combo box to a form:

form. The



Wizards"

Open

the

form

in

Design

View.

Fig.7.lO

1st

Step

of

Combo

Box

Wizard



* Depending on your choice in the first dialog box, the next options will vary. If you chose to look up values from a table or query, the following box will be displayed. Select the table or query from which the values of the combo box will come from. Click Next > and choose fields from the table or query that was selected. Click Next > to proceed.

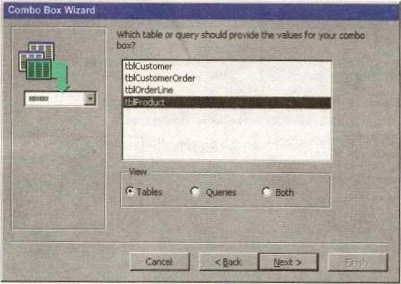


Fig-7. I I Choosing the Table in Combo Box Wizard

* On the next dialog box, set the width of the combo box by clicking and dragging the right edge of the column. Click Next

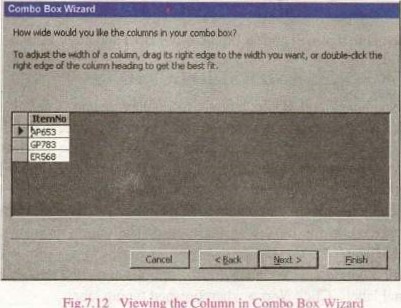


Fig.7.12

Viewing

the

Column

in

Box

108



* The next dialog box tells Access what to do with the value that is selected. Choose "'Remember the value for later use" to use the value in a macro or procedure (the value is discarded when the form is closed), or select the field that the value should be stored in. Click Next > to proceed to the final screen.

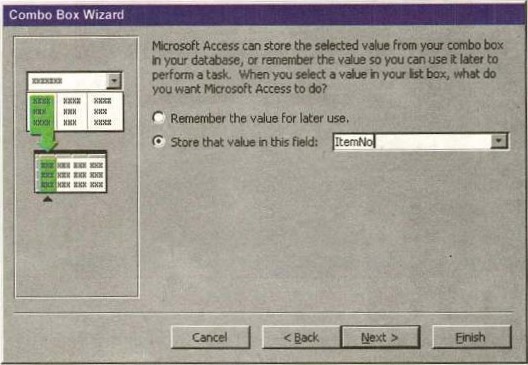


Fig.7.13 Entering data in Combo Box

* Type the name that will appear on the box's label and click Finish.

7.4 CHECK BOXES AND RADIO BUTTONS

Use check boxes and Radio buttons to display yes/no, true/false, or on/off values. Only one value from a group of radio buttons can be selected while any or all values from a check box group Can be chosen. Typically, these controls should be used when five or less option are available. Combo boxes or lists should be used for long lists Of Options. TO add a checkbox or option group:

* Click the Option Group tool on the toolbox and draw the area where the group will be placed On the form with the mouse. The option group wizard dialog box will appear.
* On the first window, enter labels for the options and click the tab key to enter additional labels. Click Next > when finished typing labels.



of

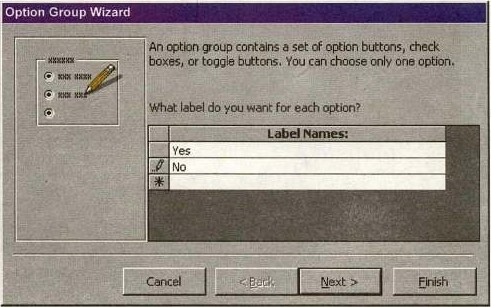
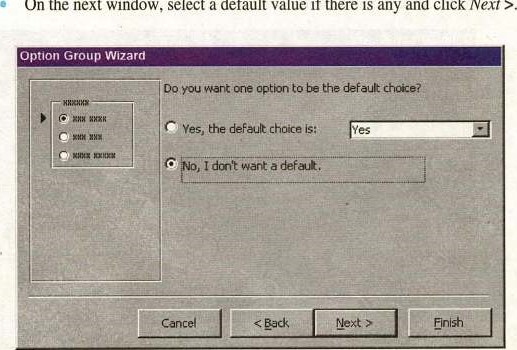


Fig. 7.14 Choosing the option in Option Group Wizard.



On

the

next

window,

select

a

default

value

if

there

is

any

and

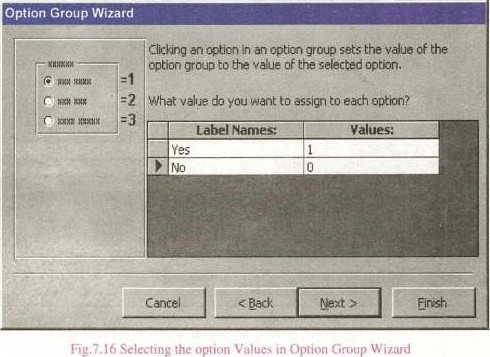
click

Next

Ftg.7.15 Defining the default option in Option Group Wizard

* Select values for the options and click Next





* Choose what should be done with the value and click Next

|  |
| --- |
| Option Group Wizard |
| You can eiher store value of a selected option in e field, or use the value later to perform a task such as printing d  What do you to do with the  Save the value for later use.  Store the yaluein &Efield: |
|
|  |

Fig.7.1 7 Storing the value of a selecled option

* Choose the type and style of the option group and click Next >.

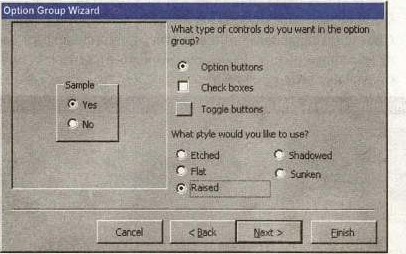


Fig.7.18 Specifying the controls and style of Option Butlon

* Type the caption for the option group and click Finish.

Command Buttons

In this example, a command button beside each record is used to open another form.

|  |  |  |
| --- | --- | --- |
| A Textbook | Com uter Science | Class 12 |

* Open the form in Design View and ensure that the Control Wizard button on the toolbox is pressed in.
* Draw the button on the form. The Command Button Wizard will then appear.
* On the first dialog window, action categories are displayed in the left list while the right list displays the actions in each category. Select an action for the command button and click Next

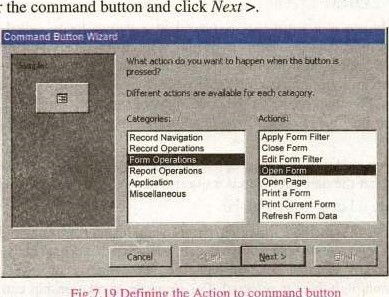


Fig:

7.19

Defining

the

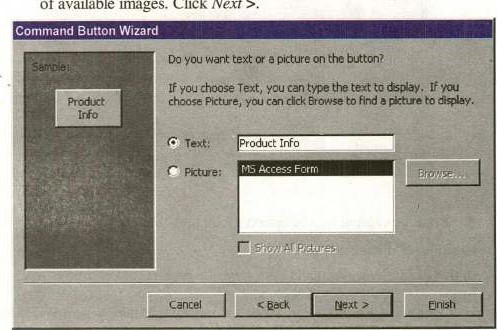
Action

command

button

* The next few pages of options will vary -based on the action you selected.

Continue selecting options for the command button.

* Choose the appearance of the button by entering text for caption of button or selecting a picture. Check the Show All Pictures box to view the full list of available images. Click Next



Class

12

fig.7.20 Finishing Command Button Wizard

* Enter a name for the command button and click Finish to create the button.

7.5 SUBFORM

A subform is a form that is placed in a parent form, called the main form. Subforms are particularly useful to display data from tables and queries that have one-to-many relationships. For example, in the sample below, data on the main form is drawn from an item information table while the subform contains all of the orders for that item. The item record is the "one" part of this one-to-many relationship while the orders are the "many" side of the relationship since many orders can be placed for the one item.

The remainder of this page explains three methods for creating subforms and they assume that the data tables and/or queries have already been created.

Create a Form and Subform

* Use this method if neither parent and child form has already been created. A main form and subform can be created automatically using the form wizard if table relationships are set properly or if a query involving multiple tables is selected. For example, a relationship can be set between

a table containing customer information and one listing customer orders so the orders for each customer are displayed together using a main form and subform. Follow these steps to create a subform within a form:

* Double-click Create form by using wizard on the database window.
* From the Tables/Queries drop-down menu, select the first table or query from which the main form will display its data. Select the fields that should appear on the form by highlighting the field names in the Available Fields list on the left and clicking the single greater symbol > button or click the double greater

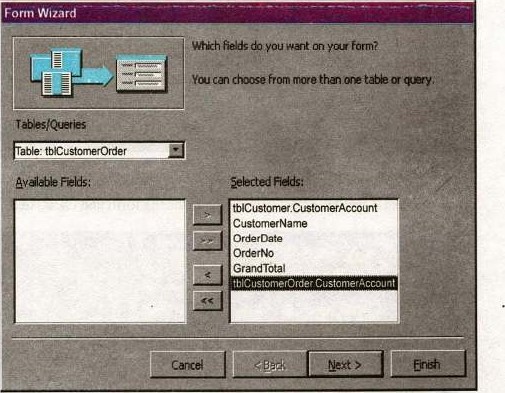


Fig.7.21 Creating Subform from Form Wizard

From the same window, select another table or query from the Tables/Queries drop-down menu and choose the fields that should appear on the form. Click Next to continue after all fields have been selected.

Choose an arrangement for the forms by selecting form with subform(s) if the fortns should appear on the same page or Linked forms if there are many controls on the main form and a subform will not fit. Click Next to proceed to the next page of options.

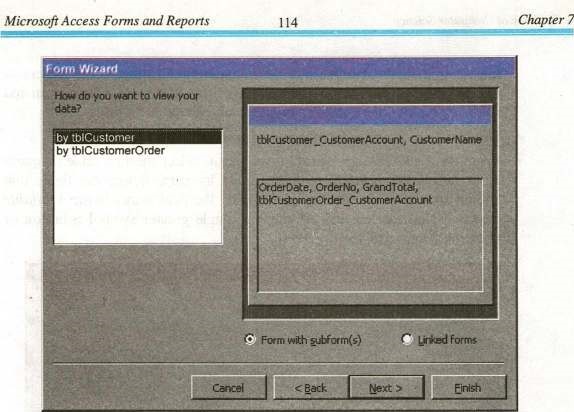


Fig.7.22 ChtRhing the Form Views

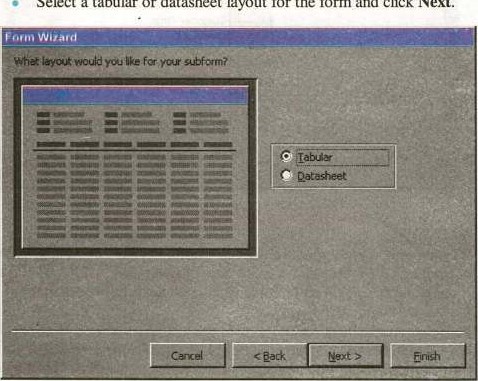
Select a tabular or datasheet layout for the form and click Next.

Fig. 7.23 Selecting the layout of subtörrn



* Select a style for the form and click Next.

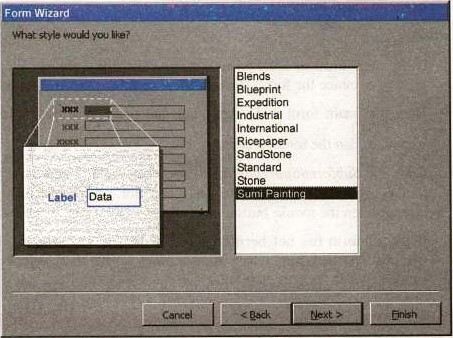


Fig. 7\_24 Speclting lhc slyle oi SubtOrm

* Enter the names for the main form and subform. Click Finish to create the forms.

Form Wizard

Fig. 7,25 Finishing the Subtorm



Cha

pter

Reports

* New records can be added to both tables and queries at once by using the new combined form.

Subform Wizard

* If the main form Or both forms already exist, the Subform Wizard can be used to combine the forms. Follow these steps to use the Subform Wizard:  Open the main form in Design View and make sure the Control Wizard button n the toolbox is pressed in.

Click the Subform/Subreport icon •Ion the toolbox and draw the outline of the subform on the main form. The Subform Wizard dialog box will appear when the mouse button is released.

* If the subform has not been created yet, select "Use existing Tables and Queries". Otherwise, select the existing form that will becorne the subform. Click Next to continue.

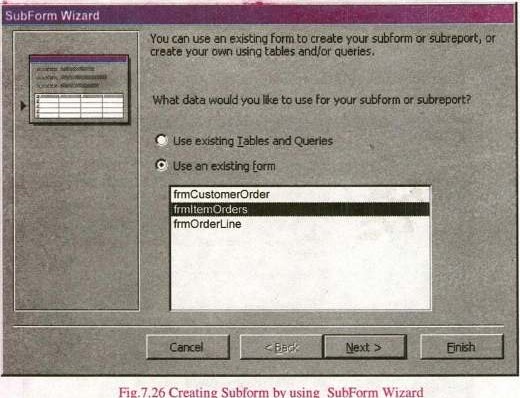


fig.

7.26

Creating

Sub-form

by

using

Wizard

* The next dialog window will display table relationships assumed by Access. Select one of these relationships or define your own and click Next.

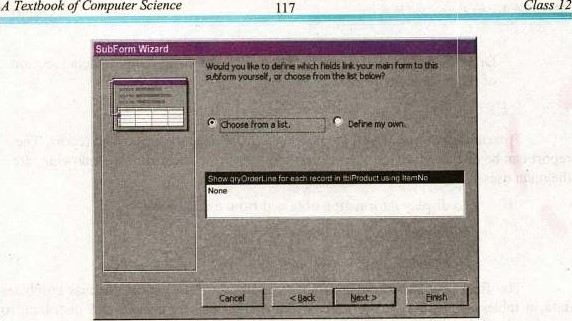


Fig.7.27 Finishing Subform by SubForm Wizard

* On the final dialog box, enter the name of the subform by choosing option 'Choose from list" option or by "define my own" and click Finish.

7.6 DR\G.AND.OROP METHOD

Use this method to create subforms from two forms that already exist. Make sure that the table relationships have already been set before proceeding with these steps.

* Open the main form in Design View and select WindowlTile Vertically to display both the database window and the form side-by-side as shown

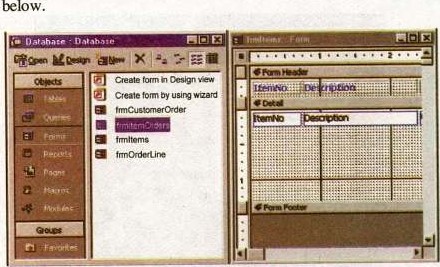


Fig.7.2S Defining Drag & Drop Methcxi



Drag the form icon beside the name of the subform onto the detail section of the main form design.

7.7 REPORTS

Presentation of processed data obtained from a database is called report. The report can be displayed on the screen, on the paper or On the disk. The following are the main uses of reports:

* To display information obtained from a database.

To display result of a query.

(iii) To produce output according to the needs of the user.

The final product of rmst database application is a report. Access combines data in tables and queries to produce a report that can be printed and distribute to people who need or request it. Reports provide means for creating printed copies of the information of database. Some reports consist of a single page, such as, order acknowledgement and invoice. Multi-page Access reports are more common than the single-page reports. These reports include catalogs, general ledgers, financial statements and examination result sheets. A report simply retrieved data from a database and presents it in a formatted form. It can retrieve data from one or Imre tables of a database. For example, in a school database, a report can be created to print the names and phone numbers of all students. The names and phone numbers of the students may store in two different tables. The report retrieved the data from the two tables and presents it in a predefined manner. Standard reports come in two basic varieties, that is, columnar and tabular.

Columnar Reports

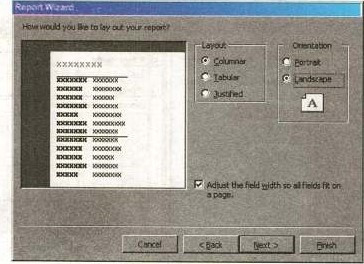


Fig. 729(a) Colummar Report

of



A

In these reports, the values of each field in each record of a table or a query are listed in one long column of text boxes. A label indicates the name Of the field and a text box to the right of the label provides the values. Columnar report spreads the information for a single record over many rows. Layout of a columnar report is shown below. It resembles the layout of a form.

Tabular Reports

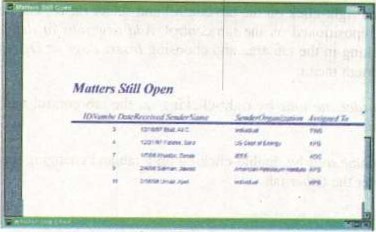


Fig.

7.294b)

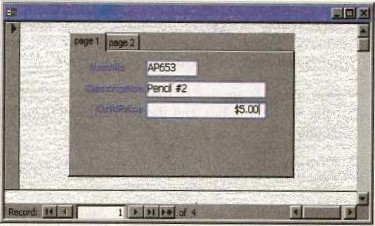
Tabular

Repon

These reports provide a column for each field of the records in rows under the column header. If you have more columns than can fit on one page, additional pages print in sequence until all columns are printed, then the next group of records is

MultiplAYage Forms Using Tabs

Tab controls allow you to easily create multi-page forms. Create a tab control by following these steps:



Fig, 7.291"' Multiple page Report

* Click the Tab Control icon on the toolbox and draw the control on the form.
* Add new controls to each tab page the same way that controls are added to regular form pages and click the tabs to change pages. Existing form controls cannot be added to the tab page by dragging and dropping. Instead, right-click on the control and select Cut from the shortcut menu. Then right-click on the tab control and select Paste. The controls can then be repositioned on the tab control. Add new tabs or delete tabs by rightclicking in the tab area and choosing Insert Page or Delete Page from the shortcut menu.
* Reorder rhe tabs by right-clicking on the tab control and selecting Page Order.
* Rename tabs by double-clicking on a tab and changing the Name property under the Other tab.

Conditional Formatting

Special formatting that depends on the control's value can be added to text boxes, lists, and combo boxes. A default value can set along with up to three conditional formats. To add conditional formatting to a control element, follow these steps:

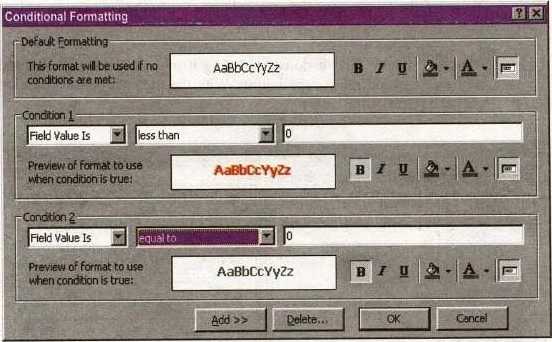
* Select the control thatthe formatting should be applied to and select Format I Conditional Formatting from the menu bar.
* Under Condition 1, select one of the following condition types:

 Field Value Is applies formatting based upon the value of the control. Select a comparison type from the second drop-down menu and enter a value in the final text box.

 Expression Is applies formatting if the expression is true. Enter value in the text box and the formatting will be added if the value matches the expression.

 Field Has Focus will apply the formatting as soon as the field has focus.

Add additional conditions by clicking the Add button and delete conditions by clicking Delete... and checking the conditions to erase.



Flg.7.30 Defining Conditional Formatting

Password Text Fields

To modify a text box so each character appears as an asterisk as the user types in the information, select the text field in Design View and click Properties. Under the Data tab, click in the Input Mask field and then click the button [...l that appears. Choose "Password" from the list of input masks and click Finish. Although the user will only see asterisks for each character that is typed, the actual characters will be saved in the database.

Change Control Type

If you decide the type of a control needs to be changed, this can be done without deleting the existing control and creating a new one although not every control type can be converted and those that can have a limited number  of types they can be converted to. To change the control type, select the control on the form in Design View and choose FormatlChange To from the menu bar. Select one of the control types that is not grayed out.

Composite Primary Keys

To select two fields for the composite primary key,

* To move the mouse over the gray column next to the field names and note that it an arrow.
* Click the mouse, hold it down, and drag it over all fields that should be primary keys and release the button. With the multiple fields highlighted, click the primary key button.

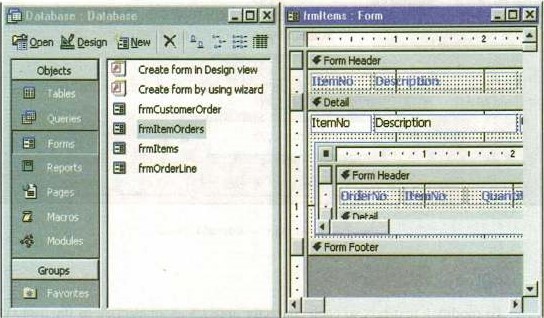


Fig.7.31 Showing composite Primary Key

Reports will organize and group the information in a table or query and provide a way to print the data in a database.

Using the Wizard

Create a report using Access' wizard by following these steps:

* Double-click the "Create report by using wizard" option on the Reports Database Window.
* Select the information source for the report by Selecting a table or query from the Tables/Queries drop-down menu. Then, select the fields that should be displayed in the report by transferring them from the Available Fields menu to the Selected Fields window using the single right greater symbol button > to move fields one at a time or the double greater symbol button to move all of the fields at once. Click the Next > button to move to the next screen.

|  |  |  |
| --- | --- | --- |
| A Textbook ofComputer Science | 23 | Class 12 |

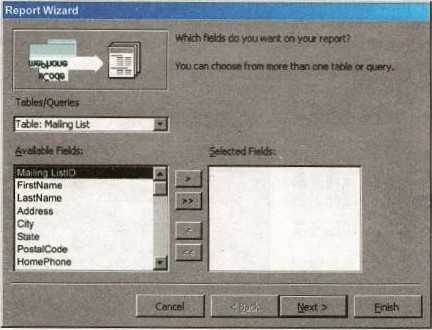


Fig.7.32 Choosing a table in Report Wizard

* Select fields from the list that the records should be grouped by and click the right greater symbol button > to add those fields to the diagram. Use the Priority buttons to change the order of the grouped fields if more than one field is selected. Click Next > to continue.

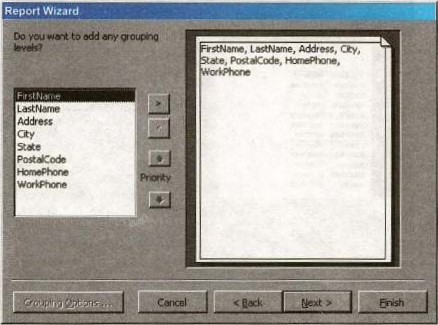


Fig.7.33.Grouping the fields in Report Wizard



* If the records should be sorted, identify a sort order here. Select the first field that records should be sorted by and click the A-Z sort button to choose from ascending or descending order. Click Next > to continue.

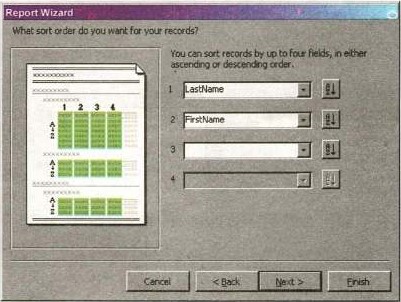


Fig.'.34 SKr-ting the record in ascending and descending ordere in Report Wizard

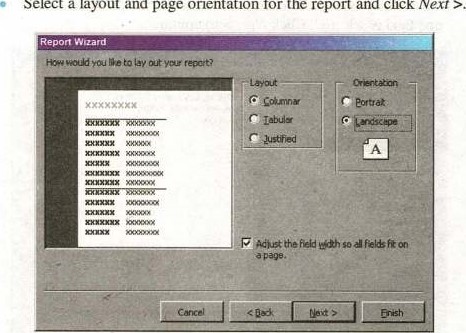
Select a layout and page orientation for the report and click Nexr

fig.7.35 Chcx»sing a Layout and paper orientation in Report Wizard

* Select a color and graphics style for the report and click Next 

A of 125





Fig.7.37 Chcx»sing a style in Report Wizard

 On the final screen, name the report and select to open it when needed. Click the Finish button to create the report.

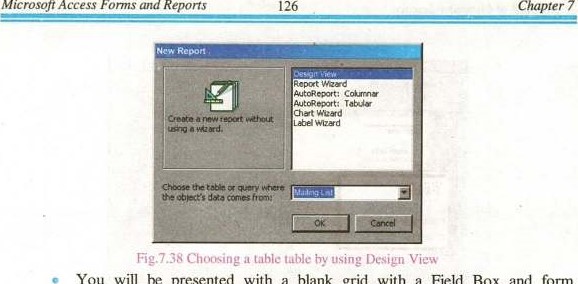


fig.7.35 Finishing in Report Wizard

('rente in Design View

TO create a report from scratch, sele€t Design View from the Reports Database Window.

New button on the Reports Database Window. Highlight 'T)esign View and choose the data source of the report from the drop-down menu  and click 0K.

You will be presented with a blank grid with a Field Box and form element toolbar that looks similar to the Design View for forms. Design the report in much the same way you would create a form. For example, double-click the title bar of the Field Box to add all of the fields to the report at once. Then, use the handles on the elements to resize them, move them to different locations, and rmdify the look of the report by using options on the formatting toolbar. Click the Print View button at the top, left corner of the screen to preview the report.

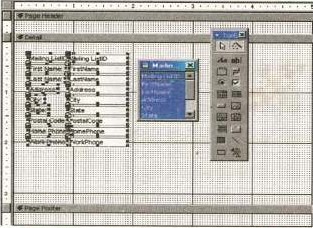


Fig.7.39 Creating report hy using Design View

Printing Reports

Select FilelPage Setup to modify the page margins, size, orientation, and column setup. After all changes have been made, print the report by selecting FilelPrint from the menu bar or click the Prim button On the toolbar.

7.8 LINKING

Unlike importing, linking objects from another database will create a link to an object in another database while not copying the table to the current database. Create a link by following these Steps:



* Open the destination database.
* Select FilelGet Externalll„ink Tables... from the menu bar.

Choose the database that the table is located in and click the Link button.

A window listing the tables in the database will then appear. Highlight the table or tables that should be linked and click OK. A link to the table will appear in the Database Window as a small table icon proceeded by a small right arrow.

This is going to be the form that will be displayed when the application starts, and will enable the user to navigate to the other parts of the database. As such, it is only going to have buttons on it. Access does have a built-in Switchboard Manager Add-ln. The switchboard form will also perform any required actions on the start-up of the database.

7.9 CREATING A SWITCHBOARD IN ACCESS

* + Open one of your databases.
  + Click on Tools, then Database Utilities, and finally Switchboard Manager.  If no Switchboard form exists, Access will display a message telling you that no Switchboard exists and will ask if you want to create one.
  + Click Yes to display the Switchboard Manager screen, and click the Edit button to edit your options.
  + From this screen, you can change the Switchboard default name from Main Switchboard to whatever you'd like.
  + Click the New button to display the Edit Switchboard Item dialog box.
  + In the Text field, type a brief description of the first item you want to add.
  + In the Command field, select the appropriate option from the drop-down list. The option you select will determine what options you'll receive in the third drop-down list.
  + For example, if you choose Open Form In Edit Mode, the Switchboard Manager will display a list of your database's forms. Choose the form and click OK.
  + Repeat this process until you've added all the items you want to the Switchboard form and click Close.

If you want to give users a second option for closing the Switchboard (besides the Close box), create a new Switchboard item named Exit and associate it with the command Exit Application; that option will close the Switchboard and the database. The final step is telling Access 2000 to run the Switchboard form when you open the database. To do so:

* + Right-click on the database window and choose Startup from the context

menu.

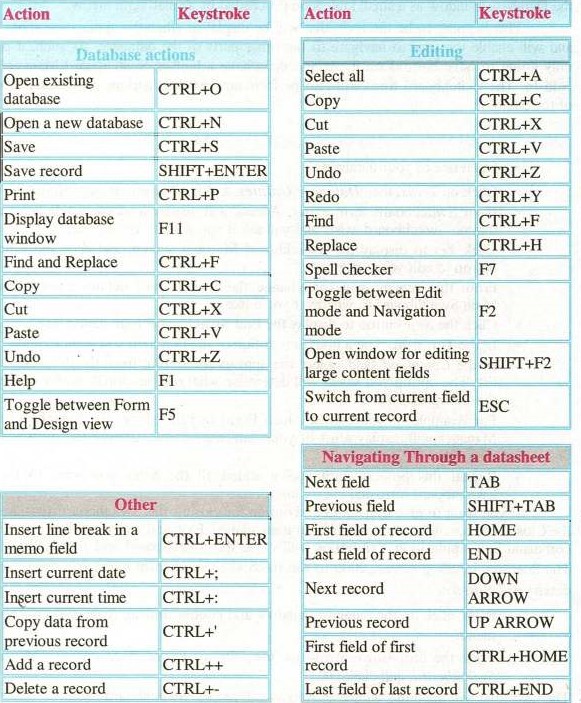
* + Click the drop-down arrow for the Display Form/ Page option, choose Switchboard, and click OK.

The next time you open this database, Access will run the Switchboard form.

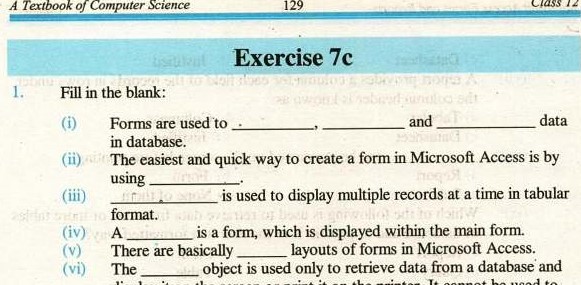


7.10 KEYBOARD SHORTCUTS 

Keyboard shortcuts can save time and the effort of constantly switching from the keyboard to the mouse to execute simple commands. Print this list of Access keyboard shortcuts and keep it by your computer for a quick reference. Note: A plus sign indicates that the keys need to be pressed at the same time.

-

object is used only to display it on the screen or print it on the edit data in a database.



Of

printer.

It

cannot

be

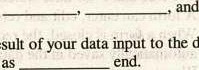
used

to

1. are can compare, summarize, and sübtotal large sets of



1. In Microsoft Access, a report can be created in any one of the three layouts. These layouts are



database

1. • Reports are the finished result of your data backed through the forms as object is used only to retrieve data from a database and displays it on the screen or print it on the printer. It cannot be used to edit data in a database. 
2. Choose the Correct Option. Forms are designed for :

a) Input Data b) Manipulate Data



 c) Accepting Change d) All of them

A(ii) A form that contains the sub form is called

a) Form  Main Form

 e) d) None of them

* 1. You can drag the bar to move the property sheet window

around on your screen.

* + 1. Title bar b) Status bar

c) Scroll bar d) , All of them

* 1. How rnany are basic layouts of forms in Micr:osoft Access.

The forms are the end of our database in Microsoft Access.

* + 1. Back end b) Front end

c) Ix)th and b None of them

(vi) Auto form displays one record at a time..

a) Tabular b) Columnar



c) Datasheet d) Justified

(vii) A report provides a column for each field of the records in rows under the column header is known as:

a) Tabular b) Columnar 

c) Datasheet d) Justified

(viii)can be previewed on the screen before printing.

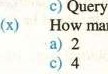
 b) 

c) Subform d) None of them



(ix) Which of the following is used to retrieve data from one or more tables of database and to present it to the user in a formatted way?

a) Report b) Form

d) Table How many are the layout of report?

d) 5

1. Write T for true and F for false statement.

A form can enter, edit and retrieve data from only One table at a time. When a form is closed, the records entered in the form are automatically saved in the database.

* 1. We cannot change any record of a table in a database using a form.

A form within another form is known as tabular form.

The tabular layout of a form is called datasheet layout.

We have not any facility to test a conditional statement in reports.

* 1. The easiest way to create a report in Microsoft Access is by using the Report Wizard.
  2. The options for layout of report are Columnar, Tabular and justified.
  3. Report are primarily used to input data easily into database tables.  The record navigation buttons at the lower-left corner of a form are used to save recently edited data in t.}r database.

1. Define the different layouts of forms ,
2. Differentiate between forms or reports
3. How is a form created by using the form visit? 7. Explain the processor for editing data through forms 8, What is sub form?
4. How is a form created for a single table?
5. What is a report?
6. How can you save and closed an auto report?



1. Define to greate a report using the report wizard ,
2. How can you apply a sort order to the report?
3. How is a report created from two tables?
4. How can report created for a single table?

of

|  |  |
| --- | --- |
|  | Chapter |

* 1. OVERVIEW

A computer is a device that follows the instructions given to it. A well-defined set of instructions given to the computer is called a computer program. A computer program is written in a programrning language. Since the emergence of computer, many programming languages have been developed but the effect of C on the computer world is everlasting. This book will remain incomplete without describing the history of the C. That's why before going into detail; let us have an overview of the history of C.

History of C

The C programming language was developed by Dennis Ritchie in 1972 at AT & T Bell Laboratories. It was derived from an earlier programming language named B. The B was developed by Ken Thompson in 1969-70 and provided the basis for the development of C. The C was originally designed to write system programs under UNIX@operating system. But over the years its power and flexibility have made it popular in industry for a vide range of applications. The earlier version of C was known as K&R (Kernighan and Ritchie) C. As the language further developed, the ANSI (American National Standards Institute) developed a standard version of the language known as ANSI C.

* 1. DEVELOPING A C PROGRAM (A STEPWISE APPROACH)

Writing a program in C is not too difficult; however it requires a good understanding of the development environment of C language. The programmer should also have the knowledge of steps required to prepare a C program for execution. 

As a first Step, install a compiler for the C language on the computer so that the source program can be compiled and •cutcd. Many compilers for C language are available from number of vendors. Any of them can be sed, but we recommend using Turbo C++.

132

82.1 Turbo C++ (A Compiler for the C language)

Turbo C++ is a Borland International's implementation of a compiler for C language. In addition to a compiler, TC provides a complete IDE (Integrated Development Environment) to Create, edit and save programs is called TC editor (Fig. 8.1). It also provides a powerful debugger that helps in detecting and removing errors in the program.

Once the TC (Turbo C) has been installed, it is very easy to write C programs in its editor. The IDE can be invoked by typing on the DOS prompt Or by double clicking the TC shortcut. The menu bar of the IDE contains menus to create, edit, compile, execute (Run) and debug a C program. A menu can be opened by either clicking the mouse on it or pressing the first highlighted character of the name Of the menu in conjunction with the Alt key. For example to open File menu, press Alt+F (hold down Alt key and then press F key).

E File Edit Semeh Run Compde I)ång Project Options Help

8.2.2 Creating and Editing a C Program

To write tl-r first C progran 01En the edit window of the Turbo IDE. This can be done by selecting FileWew option from the nenu bar. A window



appears on the screen (Fig. 8.2). This window has a double-lined border, and a cursor inside the window represents the starting point to write a program.

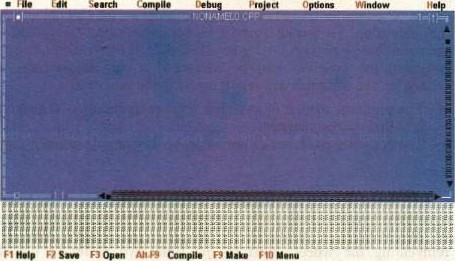


Fig. 8.2 Create. Edit and Saveu Program

We can expand this window by clicking the arrow in the upper right corner, or by selecting WindowlZoom from the menu bar. We can also navigate through the program by using the vertical and horizontal scroll bars or by using arrow keys.

8,23 Saving a C Program

After writing the C program, we should save it on the disk. This can be done by selecting FilelSave command from the menu bar or pressing the F2 key. When we select FilelSave, a dialogue box will appear. At the top of this dialogue box, there is a text box with caption Save File As. Type the name of the file in it and press the Enter key. The default path for saving the file is BIN folder. The TC assigns a default name NONAME00.cpp to the file (Fig. 8.2). To save the file in a specific folder / location with a different file name, one has to specify the absolute path.

Note:



Turbo C++ is a compiler for C++ programming language — an extension to C; Therefore it can compile programs of both C and C++. When we save program with .cpp extension, it can use many additional features that are not -supported in ANSI C. As this course is designed just for C, not C++/ therefore it is suggested to always save the programs with .c extension. When a program is saved with .c extension, the Turbo C++ compiler restricts it to only use standard features of C.

82.4 Compiling a C Program

The computer does not understand source program because instructions in the program are meaningless to the microprocessor, as it



understands only the machine language. A program that is to be executed must be in the form of machine language.

C compiler translates the source program into an object program with .0bj extension. To invoke Turbo C++ compiler, select CompilelCompile from the menu bar or press Alt + F9 key (Fig. 8.2). If there is no error in the source program, the program will be translated to object program successfully otherwise, the compiler will report errors in the program.

* The program written in any high level programming language, such as C, is called source program.
* The compiler produces an object program from the source program



* 1. Linking a C Program

 While writing a C program. the programmer may refer to many files to accomplish various tasks such as input/output etc. In case of C language, a lot of functionality is available in the form of library files. Rather than reinventing the wheel, most of the times we prefer to use the built-in functionality of the language. Such files are needed to be linked with the object file, produced by the compiler, before execution of the program.

Linking is the process in which the object file produced by the compiler is linked to many other library files by the linker. The linker is a program that combines the objcct program with additional objcct files that may be needed for the program to execute and save the final machine language program as an executable file on disk. In Turbo the linker can be invoked by selecting Compilellänk from the menu bar,

The Linker combines different library files to the object file and produces an executable file with .exe extension

* 1. Executing a C program

After successfully compiling and linking the program, we are now ready to execute it. For execution the program must be loaded into memory. This is done by the loader. Loader is a program that places executable file in memory. In Turbo C++, this is done by selecting RunlRun from the menu bar or pressing Ctrl+F9 key.

When the program is run, the screen flickers for a moment and the output Screen will disappear in a flash. To see the program's output select WindowlUser Screen or press Alt+F5. The normal DOS output screen will appear. Flowchart 8.3 describes the steps required to prepare a C program for

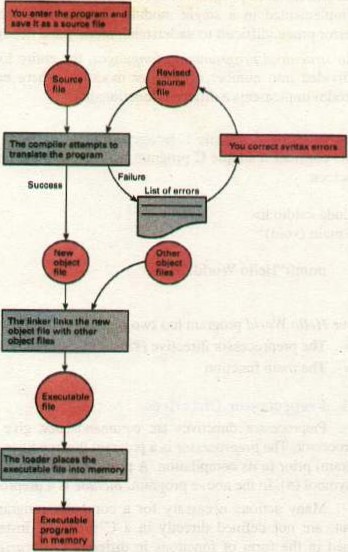
execution.

Setting the Output and Source Directories

By default, Turbo C++ places the object and executable files in the BIN subdirectory of the TC directory. This is not the right place to put these files. These files should be placed in the same directory where the source file (with -c extension) was created. To do so, select the OptionlDirectories from the

menu bar. A window appears with four fields captioned Include Directories, Library Directories, Output Directories and Source Directories. The Include

Directories filed should already be set to and Library Directories should be set to where the drive: is the drive in which the directory TC is placed. It can be C, D, or E etc. We need to set the output directory field to source file directory e.g. D:\MyPrograms etc. this is where the compiler will put .obj file and the linker will put .exe file.



8.3 BASIC STRUCTURE OF A C PROGRAM

The structure of a C program is very flexible which increases the power of the language. C is a structured programming language; therefore it provides a welldefined way of writing programs. As discussed earlier in this chapter that a C program is combined with many other files before execution. The linker does this job. But we have to specify these files to be linked. How can this be done? To answer this question and to understand the basic Structure Of the C program, We proceed with the following example:

* In unstructured programming languages, the entire logic of the program is implemented in a single module (function), which causes the program eror prone, difficult to understand, modify and debug.
* In structured programming languages, the entire logic of the program is divided into number of smaller modules, where each module (piece of code) implements a different functionality. 

Hello World — A simple C program 

Let us consider a simple C program that displays the phrase Hello World! on the screen.

#include <stdio.h> void main (void)

World!");

The above Hello World program has two parts:  The preprocessor directive (#include

* The main function

8.3.1 Preprocessor Directives

Preprocessor directives are commands that give instructions to the C preprocessor. The preprocessor is a prop-am that modifies the C program (source program) prior to its compilation. A preprocessor directive always begins with the symbol (#). In the above program, include is a preprocessor directive.

Many actions necessary for a computer program, such as input and output, are not defined directly in a C program. Instead, these actions are defined in the form of functions in different C libraries. Each library has a standard header file, which is referred to with .h extension. In the above program, the stdio.h refers to the header file containing the definition of standard input/output functions.

The include directive gives a program access to a library. This directive causes the preprocessor to insert definitions from a standard header file into a program before compilation. Hence, the statement  gives the program access to standard input and output functions.

# include Directive for Defining Identifiers from Standard Libraries

SYNTAX: headerfile>

EXAMPLE: #include <stdio.h>

#include <math.h>

The include directive tells the compiler where to find the meanings of standard identifiers (e.g., printf in the Hello World program) used in the program. These meanings are described in files called standard header files. The header file stdio.h contains information about standard input and output functions such as scanf and printf, whereas the header file math.h contains information about common mathematical functions.

Another important preprocessor directive is #define directive. It is used to define a constant macro. Examples of this macro will be discussed in subsequent chapters.

#define Directive for Defining Constant Macros

SYNTAX: #define Macro\_Name expression

 EXAMPLE: #define PI 3.142857

#define 3600

The expression may be constant, arithmetic expression or a string. C preprocessor replaces each occurrence of the identifier Macro\_Name with value of expression. The expression of the identifier Macro\_Name can not be changed during the program execution.

Constant Macro is a name that is replaced by a particular constant value before the program is sent to the compiler.

8.32 FUNCI'ION main

As shown in the above Hello World program, the definition of the main function comes next to the specification of the \*include preprocessor directive. In fact, main is the function where the execution of the C program begins. Every C program has a main function. The rest of the lines of program forms the body of the main function, the body is enclosed in braces { and }.

C programs are divided into units called functions. This division is usually done on the basis of functionality, where every function carries out a single task. However, it is no necessary to divide every program into functions. The same functionality may be achieved through a single function. But, every C program must have the functions main as the execution of the program starts from there. In this way we can say that the main function is actually the entry point of the C programs.

main Function Definition

SYNTAX: void main (void) body of mainfunction

As We know from the algebra that every function returns a single value and may accept one or more arguments (parameters). There is some resemblance between an algebraic function and the main function. The definition of the function main starts with a reserved word void. This void represents the data type of the value that is returned by the function, which means the function main returns nothing. The second void enclosed in parenthesis describes that the function main does not accept any argument. However arguments can be passed to the main function and it can also return a value. But the discussion Of this issue is out Of scope Of this book. You may find the topic in detail in many Other books.



Chapter

8

Body Of the function (enclosed in braces) consists Of C language statements, which are used to implement the program logic. There are many types of C statements that help programmers to write C programs. We shall learn much more about writing programs in C in next chapters.

8.3.3 Delimiters

Next to the function definition are braces, which indicate the beginning and end Of the function body. These braces are called delimiters. The opening brace { indicates the beginning of a block of code (set of statements) while the closing brace } represents the end of a block of code.

83.4 .Statement Terminator 

Every statement in a C program terminates with a semicolon (;). If any of the statement is missing the statement terminator, the compiler will report it the following error message.

Statement missing ;



83.5 Function printf

The last statement in the Hello World program is prinfffunction. It is used to display the output of the program on the screen. See detail in chapter 3.

8.4 COMMON PROGRAMMING ERRORS

The programmer may come across errors while writing a computer program. In programming languages, these erors are called "bugs", and the processing of finding and removing these bugs is called debugging.

When the C compiler detects an error, it displays an error message describing the cause of the error. There are three types of programming errors, these are: Syntax error, Runtime error, and Logic error.

8.4.1 Syntax Errors

A syntax error occurs when the program violates one or more grammar rules of C language. The compiler detects these errors as it attempts to translate the program. If a C statement has syntax error, it can not be translated and the program could not be executed.



There can be many causes of syntax errors, for example, missing statement terminator i.e„ the semicolon, using a variable without declaration, missing any of the delimiters i.e., { or } etc. 

8.4.2 Runtime Errors

A runtime error occurs when the program directs the computer to perform an illegal operation, such as dividing a number by zero. Runtime errors are detected and displayed by the computer during the execution of a  program. When a runtime error occurs, the computer stops executing the program and displays a diagnostic message.

8.43 Logical Errors

Logical errors occur when a program follows a faulty algorithm. The compiler can not detect logical errors; therefore no error message is reported from the compiler. Moreover, these enors don't cause the program to be crashed, that's why these are very difficult to detect. One can recognize logical errors by just looking at the wrong output of the program. Logical errors can only be detected by thorough testing of the program.

8.5 PROGRAMMING LANGUAGES

Programming languages are used to write computer programs. There are two broad categories of programming languages i.e„ low level programming languages and high level programming languages. We discuss them briefly to have an overview of both:

8.5.1 Low Level Languages

Low level languages are divided into two broad categories i.e„ machine language and assembly language. Machine language is the native language of the computer. The computer does not need any translator to understand this language. Programs written in any other language must be converted to machine language so that the computer can understand them. Every machine language instruction consists of strings of binary Os and Is. As it is very difficult for human beings to remember long sequences of Os and Is, therefore wHting programs in machine language are very difficult and error prone. So, it was thought to replace the long sequences of Os and Is in machine language with English like word. This idea provided the basis for the development of

assembly language. 

In assembly language, machine language instructions (long sequences of Os and Is) are replaced with English like words known as mnemonics (pronounced as Ne-Monics). An assembler (language translator for assembly language programs) is used to translate an assembly language programs into machine language.

8.5.2 High Level Languages

PrOgramming languages whose instmctions resemble the English language are called high level languages. Every high level language defines a set of rules for writing programs called syntax of the language. Every instruction in the high level language must confirm to its syntax. If there is a syntax error in the program, it is reported by the language translator (compiler or interpreter). The program does not translate into machine language unless the error is removed.

Common high-level languages include C, Java, Pascal, FORTRAN, BASIC, and COBOL etc. Although each of these languages was

Class n

|  |  |
| --- | --- |
| A Textbook o | rer Science |

designed for a specific purpose; all are used to write variety of application software. Some of these languages such as C and are used to write system sofiware as well. Each of these languages has some advantages and disadvantages over the other e.g., FORTRAN has very powerful mathematical  capabilities while the COBOL is ideal for writing business applications, C and C++ are very handy for writing system software while Java is equipped with strong network programming features. Besides having different features, all high level programming languages have some common characteristics are:

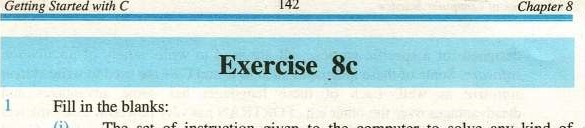
* These are English like languages, hence are close to human languages  and far from the machine language and are very easy to learn
* Programs written in high level languages are easy tb modify and debug, and more readable

These languages let the Programmers concentrate on problem being solved rather than human-machine interaction

* These describe a well defined way of writing programs

 Theée do not require a deep understanding of the machine architecture

* High level languages provide machine independence. It means programs written in a high level language can be executed on many different types of Computers with a little modification. For example, programs written in C can be executed on Intel' processors as well as Motorola processors with a little modification. 



(i) The set of instruction given to the computer to solve any kind of problem is called ANSI stands for

(iii)- The program written in high level language is known as

(iv) is a program that places the executable file in memory In —programming language , the entire logic of the program is implemented in a single module

1. are command that give instruction to C preprocessor
2. is a name that is replaced by particular constant before program is sent to the compiler
3. directive gives a program access to a library file
4. C program is divided into units, called

Every statement in a C program terminates with a

1. A language translator for Assembly language is called
2. A set of rule for writing program in high level language is known



1. Choose the correct option:  Cis a:

|  |  |
| --- | --- |
| a) High Level Language | b) Low Level Language |
| c) Assembly Language  (ii) Turbo C++ can compile: | d) Machine Language |
| a) C++ programs only | b) C and C++ programs |
| c) Turbo C programs only  (iii) Debug is the process of: | d) Turbo C++ programs only |
| a) Creating bugs in program | Identifying and removing errors |
| c) Identifying Errors | d) Removing Errors |

C was designed to write programs for:

a) Windows operating system b) Solaris operating system C) Unix operating system d) OS/2 operating system

Preprocessor directives are commands for:

 a) Microprocessor b) Language processor

c) C preprocessor d) 



(vi) The expression in define directive:

* + - * 1. can only be changed at the end of the program
        2. can not be changed



* + - * 1. can not be changed but can be redefine
        2. can not be assigned a value

(vii) Which of the following language requires no translator to execute the program:

a) c 

c) Machine language d) Assembly language

 (viii) .exe file is produced by the:

a) b) 

c) Compiler d) Interpreter



(ix) Which of the following key is used to save a file? 

a) F2 b) F3

g) F5 d) F9 

 void occupy how many bytes in memory?

 a) b) one

c) d) four

1. Write T for true and F for false statement:

The C programming language Was developed by Dennis Ritchie in 1972.

C was derived from earlier programming language named g.

* + - 1. The B was developed by Ken Thomson in 1980.
      2. The short-key for compiling a program is Alt+F9.

The compiler produces the source file from an object file.

* + - 1. The linker is a program that combines the object program with additional files.
      2. The short-key for executing the C program is Alt + F5.

 (viii) In structured programming the entire program is divided into smaller modules.

(ix) The value of a constant can be changed during the program execution.

 High level language provide machine independence.

1. Briefly describe the history of C.
2. List two reasons why it would be preferable to write a program in C rather than machine language.
3. What necessary steps taken to prepare a C program for execution? Explain with diagram.
4. Define a bug. Discuss some debugging features Of Turbo C++.
5. While writing a C program, how many types of errors can occur? Discuss briefly. Which one is the most difficult to locate and remove? Justify your
6. What is a programming language? Discuss the two main categories of programming languages.
7. Describe characteristics of high-level programming languages.
8. Briefly describe the basic Structure of a C program. 
9. How would you create, edit, compile, link and execute a C program? Discuss briefly. 
10. Differentiate the following: 

 Preprocessor Directive and the Compiler

* + - 1. Structured and Unstructured programming languages
      2. Linker and Loader