merged_topic_0: constant, symbolic, define, defines, evaluable, constants, numeric, sequence

- The definition section defines all symbolic constant
- Constant Tokens Keywords Constant Strings Operators Identifiers Special Symbol Constant is referring to fixed values that do not change during the execution of a program
- CONSTANT Numeric constant Character Constant Variable A variable is a data name that may be used to store a data value
- Defining Symbolic constant Syntax define symbolicname value of constant Eg define pi define max The following rules apply todefine statement which defines a symbolic constant
- valuevalue are constants or constant expressions (evaluable to an integer constant) and are known as case labels
- ( ) Fortran Constant A number on a string of FORTRAN characters is called a constant
- ++++
- The definition defines all the symbolic constants
- They are often referred to collectively as numeric _ type constants
- Symbolic Constants A symbolic constant is a name that substitutes for a sequence of characters
- Thus a symbolic constant allows a name to appear in place of a numeric constant a character constant or a string
- When a program is compiled each occurrence of a symbolic constant is replaced by its corresponding character sequence
- Symbolic constants are usually defined at the beginning of a program
- The symbolic constants may then appear later in the program in place of the numeric constants character constant etc that the symbolic constants represent
- A symbolic constant is defined by writing define name text where name represents a symbolic name typically written in uppercase letters and text represents the sequence of characters that is associated with the symbolic name
- What is Symbolic constant differentiate between keywords and identifier

merged_topic_1: pointer, operator, subtract, integers, operators, shift, regarded, exponential

- C operators can be classified into a number of categories
- Special operators C support some special operators of interest such as comma operator sizeof operator pointer operator (and) and member selection operator (
- Example X (int) ie is converted to integer by truncation Operator precedence and associativity Precedence is used to determine how an expression involving more than one operator is evaluated
- Example int a int pa Pointer Expression C allows us to add integers to or subtract integers from pointers as well as to subtract one pointer from another
- Pppp In addition to arithmetic operations discussed pointer can also be compared using the relational operators
- We may no use pointer in division or multiplication
- Pointer increment and scalar factor When we increment a pointer its value is increased by the length of the data type that it points to
- ++++
- C includes a number of operators which fall into several different categories such as arithmetic operators unary operators rational and logical operators assignment operators and the conditional operators bitwise operator
- Operators Arithmetic Operators There are five arithmetic operators in C They are Operators Purposes Addition Subtraction Multiplication Division remainder after integer division (also called modulus operator) There is no exponential operator in C However there is a library function (pow) to carry out exponential
- The relational operators in C are listed as Operators Meaning greater than greater than or equal to Equality equal to
- false In addition to the relational and equality operators C also includes the unary operator
- Logical operators C contains three logical operators Operator Meaning and or
- Operator is referred as logic and the operator is referred as logic or
- C contains the following five additional assignment operators and
- a f a shift shift shift g Special Operators C supports some special operators such as comma operator size of operator pointer operator (and) and member selection operators
- Write the general form of ternary operator and explain with example
- What are the various types of operators used in C language
- Describe unary binary and ternary operators with examples
- In C any nonzero value is regarded as true while zero is regarded as false

merged_topic_2: label, jump, goto, forward, backward, statement, illegible, colon

- In both the cases the control is transferred subsequently to the statementx
- The goto requires a label in order to identify the place where the branch is to be made
- The label is placed immediately before the statement where the control is to be transferred
- The general forms of goto and label statements are shown below goto label label statement label statement goto label Forward jump Backward jump Note that a goto breaks the normal sequential execution of the program
- If the label is before the statement goto label a loop will be formed and some statements will be executed repeatedly
- Such jump is known as a backward jump
- On the other hand if the label is placed after the goto label some statements will be skipped and the jump is known as a forward jump
- Another use of the goto statement is to transfer the control out of the loop (or nested loop) when certain peculiar conditions are encountered
- Avoiding goto When a goto is used many compilers generate a less efficient code
- Jump Statement The jump statement unconditionally transfers program control one point to another point in a program
- ++++
- The goto statement transfers the control to the labeled statement somewhere in the current function
- The general syntax of goto statement goto label label statement Here label is any valid C identifier and it is followed by a colon
- Whenever the statement goto label is encountered the control is transferred to the statement that is immediately after the label
- Generally the use of goto statement is avoided as it makes program illegible and unreliable

merged_topic_3: satisfied, loop, body, looping, condition, executed, statements

- Decision making and looping Looping (or iteration) is the process of executing a

sequence of statements until some condition for termination of loop is satisfied
- If the conditions are not satisfied then the body of the loop will not be executed
- If the condition is true the program continues to evaluate the body of the loop once again
- If the condition is true the body of loop is executed otherwise the loop is terminated and execution continues with the statement that immediately follows the loop
- If the condition is satisfied the body of the loop is again executed
- ++++
- This ensures that the algorithm will ultimately terminate
- So a loop may be defined as a block of statements which are repeatedly executed for a certain number of times or until a particular condition is satisfied
- The control statement in loop decides whether the body is to be executed or not
- After the execution again the condition is checked and if it is found to be true then again the statements in the body of loop are executed
- The body of this loop may contain a single statement or a block of statements
- If the condition is true then again the loop body is executed and this process continues until the condition becomes false
- Instead the remaining loop statements are skipped and the computation proceeds directly to the next pass through the loop

merged_topic_4: getchar, character, formatted, putchar, input, getche, putch

- Managing input and output operation Reading a character Reading a single character can be done by using the function getchar
- The getchar takes the following form variable_namegetchar ( ) Variable_name is a valid C name that has been declared as char type
- When this statement is encountered the computer waits until a key is pressed and then assigns this character as a value to getchar function
- Since getchar is used on the right hand side of an assignment statement the character value of getchar is in turn assigned to the variable name on the left
- For example char name namegetchar ( ) The getchar ( ) function accept any character keyed in
- This could create problem when we use getchar ( ) in a loop interactively

- Writing a character Like a getchar there is a analogous function putchar for writing characters one at a time to the terminal
- It takes the form as shown below putchar (variable_name) where variable_name is a type char variable containing a character
- For example answerY putchar (answer) this statement display the character Y include include void main ( ) char name char ch int i printf (nEnter the name) while ((chgetchar ( )) n) nameich i namei printf (nThe name is s name) getch ( ) Formatted input Formatted input refers to an input data that has been arranged in particular format
- For example scanf (f f f x y z) Inputting Character Strings Scanf can input strings containing more than one character
- Following are the specifications for reading character strings ws or wc Formatted output The printf statement provides certain features that can be effectively exploited to control the alignment and spacing of printout on the terminals
- Characters that will be printed on the screen as they appear
- Reading line of text scanf with s or ws can read only string without whitespaces
- that can be used to read a line containing a variety of characters including whitespace
- For example char name scanf (nname) printf (sname) Note we can use getchar and gets functions to take input to character array variable
- For example to getchar look at unformatted input example
- ++++
- getchar ( ) and putchar ( ) The getchar ( ) function reads a character from a standard input device
- The general syntax is character_variable getchar ( ) where character_variable is a valid C char type variable
- When this statement is encounted the computer waits until a key is pressed and assign this character to character_variable
- The putchar ( ) function displays a character to the standard output device
- The general syntax of putchar ( ) function is putchar (character_variable) where character_variable is a char type variable containing a character
- getch ( ) getche ( ) and putch ( ) The functions getch ( ) and getche ( ) reads a single character the instant it is typed without waiting for the enter key to be hit
- The difference between them is that getch ( ) reads the character typed without echoing it on the screen while getche ( ) reads the character and echoes (displays) it on the screen

- The general syntax of getch ( ) character_variable getch ( ) Similarly the syntax of getch ( ) is character_variable getche ( ) The putch ( ) function prints a character onto the screen
- The general syntax is putch (character_variable) These three functions are defined under the standard library function conioh and hence we should include this in our program using the instruction include include main ( ) char ch ch clrscr ( ) printf (Enter first character) ch getch ( ) printf (n Enter second character) ch getche ( ) printf (n First character) putch (ch) printf (n Second character) putch (ch) Output Enter first character Enter second character b First character a Second character b Since the first input is taken using getch ( ) function the character a entered is not echoed
- However using getche ( ) function we can see what we have typed
- The last getch ( ) simply takes a character but doesnot store it anywhere
- gets ( ) and puts ( ) The gets ( ) function is used to read a string of text containing whitespaces until a newline character is encountered
- The general syntax of gets ( ) is gets (string_variable) The puts ( ) function is used to display the string onto the terminal
- The general syntax of puts ( ) is puts (string _variable) This prints the string value of string_variable and then moves the cursor to the beginning of the next line on the screen
- (PU) Distinguish between getc ( ) a getchar ( )

merged_topic_5: marks, student, total, float, highest, grade, roll, remarks

- One dimensional array include include include float mean (float aint size) int i float sum for (iisizei) sumai return (sumsize) float star (float aint size) int i float xmstd xmmean (asize) for (iisizei) std (xmai) (xmai) stdsqrt (stdsize) return std void main ( ) float a i printf (nEnter element) for (iii) scanf (fai) printf (n The standard deviation is fstar (a)) getch ( ) Passing Two dimensional Array to the function The rules are
- of a student and marks obtained by him in subjectsDeclare array to hold the data of students
- Pass this to a function that displays the marks of student who has a highest total marks
- include include struct student int roll int marks int total void input (struct student s) int ij for (iii) printf (nEnter the roll number of student) scanf (dsiroll) printf (nEnter

marks for subjects) sitotal for (jjj) scanf (dsimarksj) sitotalsitotalsimarksj void display (struct student s) int ilocationmax maxstotal location for (iii) if (maxsitotal) serching highest marks location maxsitotal locationi printf (nRecord of student who score highest marks) printf (nRoll number of studentdslocationroll) printf (nEnter marks for subjects) for (iii) printf (nMarks in d subjectdislocationmarksi) void main ( ) struct student civil input (civil) display (civil) getch ( ) Create a structure STUDENT containing name symbol number name of subjects mark of each subject and total mar as its members

- Write a program that uses this structure and reads data for a student and gives the total marks as the output

- Refer to above program and display only total marks of all the student Write a program to compute any two instant of distances in a format feetinches using structure

- ++++

- The method of assuming grade is as per grade A per grade B per grade C per grade D per main ( ) float m m m m total per char grade clrscr ( ) printf (Enter marks of subjects) scanf (fffffm m m m) total mmmm per total if (per) grade A elseif (per) grade B elseif (per) grade C elseif (per) grade D else grade F printf (Percentage is fn Grade is cn per grade) getch ( ) Equivalent code in simple if statement if (per) grade A if (per) grade B if (per) if (per) grade F In else_if ladder whenever a condition is found true other conditions are not checked while in if statement all the conditions will always be checked wasting a lot of time and moreover the conditions here are more lengthy

- M Singh) printf (n Names name) name (char) realloc (ame) strcpy (namecaptain B M Singh) pritf (nNamesname) getch ( ) output Name B M Singh Name Captain B M Singh Program to read marks obtained by ditstudent calculate sum average using pointer include include include main ( ) int n i float p sum avg clrscr ( ) printf (n How Many Students aretheret) scanf (dn) printf (n Enter marks of each studentsn) p (float) malloc (nsizeof (float)) for (i in i) scanf (f (pi)) avgsumn printf (The average marks of) for (i in i) printf (f t (p i)) printf (f t is avg) free (P) getch ( ) output How many students are there

- Enter marks of each student The average marks of is write a pg to read an array of n integers using dynamic memory allocation and display the largest and smallest element int main ( ) int n i intnummaxmin clrscr ( ) printf (n enter number of elements in our array) scanf (dn) num (int) calloc (nsizeof (int)) printf (nEnterd integersn) for (i i (numi)) min (numi) printf (n the maximum numberdmax) printf (n the minimum numberdmin) getch ( ) output Enter numbers of element in our array Enter integers

The maximum number The minimum number Some Important Questions What are the relationship between arrays and pointer

- WAP using structure to read and display the data entered by the user include main ( ) struct student char name int roll float marks char remark struct student s clrsc ( ) printf (enter name t) gets (sname) printf (In enter rollt) scanf (d sroll) printf (n enter marks t) scanf (f smarks) printf (enter remarks p for pass or f for fail t) sremark getch ( ) printf (nn The students information is n) printf (Student Name ItItIt Roll It Marks It Remarks) printf (n n) printf (sttdtftc sname sroll smarks sremark) getch ( ) output enter name Ram Singh enter roll Enter remark p for pass or f for fail P The students information is Student Name Roll Marks Remarks Ram Singh P Array of Structure Like array of int float or char type there may be array of structure

- Use this structure to read and display records of student main ( ) struct student charname int roll float marks char remark struct student st int i clrscr ( ) for (i i i) printf (n Enter Information of student Nodn i) printf (Name t) scanf (s siname) printf (n Roll t) scanf (d siroll) printf (n Markst) scanf (f simarks) printf (remark (pf) t) siremark getch ( ) printf (nn The Detail Information is n) printf (Student Name t Roll t Marks t Remarks) printf (n_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ n) for (i i i) printf (sttdtf tcnsiname siroll simarks siremark) getch ( ) output Enter Information of student No Name Ram Roll Marks Remarks P Enter Information of Student No Name Shyam Roll Marks Remarks P Enter Information of Student No Name Bikash Roll Marks Remarks P Enter Information of Student No Name Jaya Roll Marks Remarks P Enter Information of Student No Name Nisha Marks Remarks F The Detail Information is Student Name Roll Marks Remarks Ram P Shyam P Bikash P Jaya P Nisha F Initializing array of structure Array of structure can be initialized in the same way as a single structure

- Create a union named student that has roll and marks as member one at a time and display the result one at a time void main ( ) union student int roll float marks union student st stroll printf (nRolld stroll) stmarks printf (nMarksf stmarks) Roll Marks If two members are used simultaneously the output is unexpected as following

- void main ( ) union student introll float marks union student st stroll stmarks printf (nRolld stroll) printf (nMarksf stmarks) Output Roll Marks Where roll is zero as memory is replaced by another variable stmarks

- EOF) fputc (ch fdest) printf (Sucessfully copied) fclose (fsource) fclose (fdest) getch ( ) Program to understand fprintf ( ) include Struct student char name float marks std main ( ) FILE fp int i n fp fopen (studentdat w) printf (enter number of records) scanf (d n) for (i i char name float marks std main ( ) FILE fp fp fope (studentdat

- Write an array of structure to a life then read its content to display to the screen Void main ( ) struct student Char name int roll float marks struct student s st int i float tempMarks FILE fptr fptr fopen (dstudenttxt

merged_topic_6: constant, real, point, fractional, mantissa, base, constants, integer, rarely, decimal

- Number are called numeric constant
- etc are some numeric constants
- There are two types of Numeric Constant
- Real Constant INTEGER Constant Integer written without decimal point is called fixed point constant or integer constants
- No other character should occur in fixed point constant
- Real Constant Any number written with one decimal point is called a floating point constant of real constant
- A real constant can be expressed in any one of the following two forms i) Fractional form ii) Exponential form The following rules apply for the real constant in fractional form
- A real constant is written in the decimal form with the digitals and the decimal point
- No special symbols such as etc are allowed in a real constant
- The general form of the exponential floating point constant is Mantissa E exponent
- The mantissa must be valid real constant in fractional form
- All the rules of the fractional form applied to the mantissa
- ++++
- Integer and floating point constants represent numbers
- Integer Constants An integer constant is an integervalued number
- Integer (number) constants can be written in three different number systems decimal (base) octal (base) and hexadecimal (base)
- Beginning programmers rarely however use anything other than decimal integer constants

merged_topic_7: testcondition, body, continues, loop, process, executed, condition, statements

- Body of the loop
- The testcondition is evaluated and if the condition is true then the body of the loop is executed
- After execution of the body the test condition is once again evaluated and if it is true the body is executed once again
- This process of repeated execution of the body continues until the testcondition finally becomes false and the control is transferred out of the loop
- Body of the loop Test condition
- This process continues as long as the condition is true
- Since the testcondition is evaluated at the bottom of the loop the dowhile construct provides an exitcontrolled loop and therefore the body of the loop is always executed at least once
- The value of the control variable is tested using the testcondition
- This process continues till the value of the control variable fails to satisfy the testcondition
- ++++
- This ensures that the algorithm will ultimately terminate
- So a loop may be defined as a block of statements which are repeatedly executed for a certain number of times or until a particular condition is satisfied
- The control statement in loop decides whether the body is to be executed or not
- After the execution again the condition is checked and if it is found to be true then again the statements in the body of loop are executed
- The body of this loop may contain a single statement or a block of statements
- If the condition is true then again the loop body is executed and this process continues until the condition becomes false
- Instead the remaining loop statements are skipped and the computation proceeds directly to the next pass through the loop

merged_topic_8: constant, real, point, fractional, mantissa, unsigned, long, floating, appending, constants

- Number are called numeric constant
- etc are some numeric constants
- There are two types of Numeric Constant

- Real Constant INTEGER Constant Integer written without decimal point is called fixed point constant or integer constants
- No other character should occur in fixed point constant
- Real Constant Any number written with one decimal point is called a floating point constant of real constant
- A real constant can be expressed in any one of the following two forms i) Fractional form ii) Exponential form The following rules apply for the real constant in fractional form
- A real constant is written in the decimal form with the digitals and the decimal point
- No special symbols such as etc are allowed in a real constant
- The general form of the exponential floating point constant is Mantissa E exponent
- The mantissa must be valid real constant in fractional form
- All the rules of the fractional form applied to the mantissa
- ++++
- An unsigned integer constant can be identified by appending the letter ( ) (either upper or lowercase) to the end of the constant
- Long integer constants may exceed the magnitude of ordinary integer constants but require more memory within the computer
- A long integer constant can be identified by appending the letter L (either upper or lowercase) to the end of the constant
- An unsigned long integer may be specified by appending the letters UL to the end of the constant
- However the U must preced the L Several unsigned and long integer constants are shown below Constant Number System U decimal (unsigned) L decimal (long) UL decimal (unsigned long) L octal (long) U octal (unsigned) XFFFFFUL hexadecimal (unsigned long) Floating Point Constants A floating point constant is a base number that contains either a decimal point or an exponent (or both)
- Several valid floating point constants
- E e e The following are not valid floating point constants for the reason stated
- The quantity can be represented in C by any of the following floating point constants

merged_topic_9: content, mode, file, existing, opens, exist, new, erased

- File opening Modes The different types of file opening modes are r This mode open file for reading only

- w This mode open file for writing only
- a This mode open file for appending (or adding) data to it
- When the file is opened in this mode the file is opened with the current content safe
- r This mode opens the file for reading existing content writing new contents and modifying existing content of the file
- w This mode opens the file for writing new content reading them back and modifying the existing content of the file
- a This mode opens the file for appending new content to the end of file reading existing content from the file But can not modify existing contents
- Attempting to write a writeprotected file
- ++++
- read write append etc)
- w (write) If the file dosent exist then this mode creates a new file for writing and if the file already exists then the previous data is erased and the new data entered is written to the file
- a (append) If the file doesnt exist then this mode creates a new file and if the file already exists then the new data entered is appended at the new data entered is appended at the end of existing data
- In this mode the data existing in the file is not erased as in w mode
- r (read) This mode is used for opening an existing file for reading purpose only
- The file to be opened must exist and the previous data of file is not erased
- w (write read) This mode is same as w mode but in this mode we can also read and modify the data
- If the file doesnt exist then a new file is created and if the file exists then previous data is erased
- r (read write) This mode is same as r mode but in this mode we can also write and modify existing data
- Since we can add new data and modify existing data so this mode is also called update mode
- at (append read) This mode is same as the a mode but in this mode we can also read the data stored in the file
- If the file doesnt exist a new file is created and if the file already exists then new data is appended at the end of existing data in this mode

merged_topic_10: section, closing, brace, executable, appear, userdefined,

prototypes, ahead, function

- This means that we can execute them directly but can not read or modify them
- This section also declares all the userdefined functions
- Every C program must have one main ( ) function section
- This section contains two parts declaration part and executable part
- The declaration part declares all the variables used in the executable part
- There is at least one statement in the executable part
- These two parts must appear between the opening and the closing braces
- The program executing begins at the opening brace and ends at the closing brace
- The closing brace of the main function section is the logical end of the program
- All statements in the declaration and executable parts end with a semicolon ( )
- The subprogram section contains all the userdefined functions that are called in the main function
- Userdefined functions are generally placed immediately after the main function although they may appear in any order
- All section except the main function section may be absent when they are not required
- define statements must not end with a semicolon
- define statements may appear anywhere in the program but before it is referenced in the program
- There is no need to put braces around these blocks
- The unit can be invoked from other part of the program
- Such declaration is available for all the function in the program
- Function Definition A function must be defined before it is used anywhere in the program
- add ( ) When the compiler encounters the function call the control is transferred to the function definition
- And it executes each line of code in function and returns the control to the main program again
- A function may be used by many other program
- This is a nonexecutable statement
- ++++
- One of the function must be called main ( ) A function is a subroutine that may include

one or more statements designed to perform a specific task

- Every C program must have one main ( ) function section
- The subprogram section contains all the userdefined functions that are called in the main ( ) function
- Userdefined functions are generally placed immediately after the main ( ) function although they may appear in any order
- All section except the main ( ) function section may be absent when they are not required
- The function main ( ) is always present in each program which is executed first and other functions are optional
- The function main ( ) is an user defined function except that the name of function is defined or fixed by the language
- This function is executed first when the program starts execution
- Function Declaration or Prototype The function declaration or prototype is model or blueprint of the function
- If functions are used before they are defined then function declaration or prototype is necessary
- Many programmers prefer a topdown approach in which main appears ahead of the programmer defined function definition
- Function prototypes are usually written at the beginning of a program ahead of any userdefined function including main ( )
- Function prototypes provide the following information to the compiler
- In bottomup approach where userdefined functions are defined ahead of main ( ) function there is no need of function prototypes

merged_topic_11: function, calling, return, does, receive, keyword, argument

- Void data type (for empty set of value and nonreturning functions) the void type specifies an empty set of values
- It is used as the return type for functions that do not return a value
- Function with no arguments and no return values When a function has no argument it does not receive any data from the calling function
- Similarly when it does not return value the calling function does not receive any data

from the called function
- Function with no argument and return type In this function the function does not take any input from the calling function but it returns value to the calling function
- Recursive A function is said to be recursive if a statement in the body of the function calls itself
- The return statement is necessary only when the function is returning some data back to the calling function
- ++++
- If there is no return value the keyword void is used
- There are two ways in which it can be used return return (expression) where return is a keyword
- In this case only return keyword is written
- The value returned by the return statement may be any constant variable expression or even any other function call which returns a value
- Similarly when called function does not return a value the calling function does not receive any data from it
- There is no argument within parenthesis which implies function has no argument and it doesnt receive any data from the called function

merged_topic_12: position, current, offset, fseek, file, putw, syntax, rewind

- ftell takes a file pointer and return a number of type long that corresponds to the current position
- This function is useful in saving the current position of a file which can be used later in the program
- It takes the following form N fte (fp) n would give the relative offset (in bytes) if the current position
- This function helps us in reading a file more than once without having to close and open the file
- fseek function is used to move the file position to a desired location within the file
- It takes the following form fseek (file_ptroffsetposition) file_ptr is a pointer to the file concerned offset is a number or variable of type long and position is an integer number
- the offset specifies the number of position (byte) to be moved from the location specified by position
- The position can take one of the following three values value meaning Beginning of

file Current position End of file Example fseek (fpL) meaning go to begining refer to class note for example FORTAN character set The following is the set of character used in FORTRAN

- ++++

- Its syntax is char_variable fget (file_ptr_variable) fputc ( ) It is used to write a character to a file

- Its syntax is fputc (char_variable file_ptr vriable) b) String IO functions fgets ( ) It is used to read string from file

- Its syntax is fgets (string int_value file_ptr_variable) fputs ( ) It is used to write a string to a file

- Its syntax is fputs (string file_ptr_variable)

- Its syntax is fprintf (file_ptr_variablecontrol string list variables) fscanf ( ) This function is used to read some integer float char or string froma file

- Its syntax is fscanf (file_ptr_variablecontrol string list_variables) Creating data files Some sample programs create a file named testtxt and write some text Welcome to Eastern College of Engineering to the file

- File Pointer Device Stdin Standard input device (keyboard) Stdout Standard OUTPUT devices (screen) Stderr Standard error OUTPUT device (screen) Some other unformatted function Intger IO getw ( ) putw ( ) for binary mode Record IO fread ( ) fwrite Putw ( ) used to write integer value th file

- Their syntaxes are int putw (int value FILE fpir) int getw (FILE fptr) fwrite ( ) used for writing an etire block to a given file

- Their syntax are fwrite (ptr size_of_array_or_structure number_of_array_or_structure fptr) fread (ptr size_of _array_or_structure number_of_structure_or_array fptr) Random Acess to File We can acess the data stored in the file in two ways sequently or random

- The syntax is int fseek (FILE fp long displacement int origin) where fp is file pointer displacement is long integer which can be positive or negative and it denotes the number of bytes which are skipped backward (if negative) or forward (if positive) from the position specified in the third argument

- It can take one of these three values Constant Value Poition SEEK_SET Beginning of file SEEK_CURRENT Current position SEEK_END End of File rewind ( ) This function is used to move the file position positive to the beginning of the file syntax rewind (File fp) using rewind (fp) is equivalent to fseek (fp OL O) ftell ( ) This function returns the current position of the file position pointer

merged_topic_13: programmer, operating, crashing, conjunction, thorough, algebraic, augmented, assisting, visual, road

- It is a set of instructions that tells the computer what to do when the computer operator does something
- It operates in close conjunction with operating system and enables us (the programmer) to exploit certain capabilities of operating system while creating the program
- It requires a thorough knowledge (low level) of the hardware for which the program is being created
- If not it will inform the programmer where rules have been violated
- It should be able to detect unreasonable or error conditions and indicate them to the programmer or user without stopping all operations crashing the system
- It facilitates topdown modular programming
- It is easy to locate and isolate a faulty function for further investigations
- ++++
- It is simply a method of assisting the program to lay out in a visual two dimensional format ideas on how to organize a sequence of steps necessary to solve a problem by a computer
- It is basically the plan to be followed when a program is written
- It acts like a road map for a programmer and guides himher how to go from starting point to the final point while writing a computer program
- It displays the dynamics of a program and allows us to examine and compare the information at various points
- Its instructions consists of terms that resemble algebraic expression augmented by certain English keywords such as if else for do and while etc

merged_topic_14: file, opening, data, defined, secondary, files, level, high, buffer, oriented

- The basic file operations performed are Naming a file Opening a file Reading data from a file Writing data from a file Closing a file
- Defining and opening a file If we want to perform operation in a file in the secondary memory we must specify certain things about the file to the operating system

- Example Studenttxt Employdat Data structure of a file is defined as FILE in the library of standard IO function definitions
- Therefore all files should be declared as type FILE before they are used
- FILE is a defined data type
- For example we may write data to the file or read the already existing data
- Following is the general format for declaring and opening a file FILE fp fp fopen (filenamemode) the first statement declares the variable fp as a pointer to the data type FILE
- The second statement opens the file named filename and assigns an identifier to the FILE type pointer fp
- ++++
- Programming language C has various library functions for creating and processing data files
- High level (standard or stream oriented) files
- Low level (system oriented) files
- In high level data files the available library functions do their own buffer management where as the programmer should do it explicitely in case of lower level files
- For each binary and text files there are a number of formatted and unformatted library functions in C fig classification of files Data files High Low Text Binary Opening and closing a data file Before a program can write to a file or read from a file the program must open it
- While working with high level data file we need buffer area where information is stored temporarily in the course of transferring data between computer memory and data file
- A structure named FILE is defined in the file stdioh that contains all information about the file like name status buffer size current position and of file status etc
- (PU) What is a files ad explain its importance in C programming
- What is a file

merged_topic_15: prototype, declaration, local, function, global, parameter, type, returned, mul, return

- Function declaration or Function prototype
- Function Prototype A function prototype is a declaration of the function that tells the program about the type of the value returned by the function and the number and type

of each argument

- Functiontype functionname (parameter list) Example void sum (int a int b) when we place the declaration above all the function (in the global declaration section) the prototype is referred to as global prototype
- When we place it in a function definition (in the local declaration section) the prototype is called a local prototype
- The prototype declaration should be similar to the function header
- ++++
- The general syntax of function prototype is return_type function_name (type type typen) where return_type specifies the data type of the value returned by the function
- A function can return value of any data type
- The general syntax is return_type function_name (parameter parameter parametern) statements where return_type is the data type specifier of data returned by the function
- The syntax of teh call if function return_type is void is function_name (parameter name) If function return int float or any other type value then we have to assign the call to same type value like variable function_name (parameter) For example m mul (x y) The type of m is same as the return type of mul function
- This type of function is defined as void function_name ( ) body of function The keyword void means the function does not return any value

merged_topic_16: file, opening, data, defined, secondary, position, putw, syntax, rewind

- The basic file operations performed are Naming a file Opening a file Reading data from a file Writing data from a file Closing a file
- Defining and opening a file If we want to perform operation in a file in the secondary memory we must specify certain things about the file to the operating system
- Example Studenttxt Employdat Data structure of a file is defined as FILE in the library of standard IO function definitions
- Therefore all files should be declared as type FILE before they are used
- FILE is a defined data type
- For example we may write data to the file or read the already existing data
- Following is the general format for declaring and opening a file FILE fp fp fopen (

filenamemode) the first statement declares the variable fp as a pointer to the data type FILE

- The second statement opens the file named filename and assigns an identifier to the FILE type pointer fp

- ++++

- Its syntax is char_variable fget (file_ptr_variable) fputc ( ) It is used to write a character to a file

- Its syntax is fputc (char_variable file_ptr vriable) b) String IO functions fgets ( ) It is used to read string from file

- Its syntax is fgets (string int_value file_ptr_variable) fputs ( ) It is used to write a string to a file

- Its syntax is fputs (string file_ptr_variable)

- Its syntax is fprintf (file_ptr_variablecontrol string list variables) fscanf ( ) This function is used to read some integer float char or string froma file

- Its syntax is fscanf (file_ptr_variablecontrol string list_variables) Creating data files Some sample programs create a file named testtxt and write some text Welcome to Eastern College of Engineering to the file

- File Pointer Device Stdin Standard input device (keyboard) Stdout Standard OUTPUT devices (screen) Stderr Standard error OUTPUT device (screen) Some other unformatted function Intger IO getw ( ) putw ( ) for binary mode Record IO fread ( ) fwrite Putw ( ) used to write integer value th file

- Their syntaxes are int putw (int value FILE fpir) int getw (FILE fptr) fwrite ( ) used for writing an etire block to a given file

- Their syntax are fwrite (ptr size_of_array_or_structure number_of_array_or_structure fptr) fread (ptr size_of _array_or_structure number_of_structure_or_array fptr) Random Acess to File We can acess the data stored in the file in two ways sequently or random

- The syntax is int fseek (FILE fp long displacement int origin) where fp is file pointer displacement is long integer which can be positive or negative and it denotes the number of bytes which are skipped backward (if negative) or forward (if positive) from the position specified in the third argument

- It can take one of these three values Constant Value Poition SEEK_SET Beginning of file SEEK_CURRENT Current position SEEK_END End of File rewind ( ) This function is used to move the file position positive to the beginning of the file syntax rewind (File fp) using rewind (fp) is equivalent to fseek (fp OL O) ftell ( ) This function returns the current position of the file position pointer

merged_topic_17: matrix, print, enter, column, array, second, row

- Store the sum of A and B in C
- Practice Write to program to arrange list of elements in ascending order include include include void main ( ) int ntemp int ij printf (nEnter the list of elements) for (iinj) tempni ninj njtemp printf (n The elements in assending order) for (iii) printf (ndni) getch ( ) Two Dimensional Array A twodimensional array is an array in which each element is itself an array
- Practice Write a program that adds the individual rows of a two dimensional array of m by n and store the sums of rows into a single dimensional array using functions
- Write a function that takes a two dimensional array and onedimensional array and process the result and store in onedimensional array
- Matrix Multiplication include include void input (int aint rint c) int ij for (iiri) for (jjcj) scanf (daij) void display (int aint rint c) int ij for (iiri) for (jjcj) printf (daij) printf (n) void multiply (int aint bint cint rint cint c) int ijk for (iiri) for (jjcj) cij for (kkck) cijaikbkj void main ( ) int abcrrcc clrscr ( ) printf (nEnter the row and column for first matrix) scanf (ddrc) printf (nEnter the row and column for second matrix) scanf (ddrc) if (cr) printf (nMultipication is not possible) else printf (nenter the value to first matrix) input (arc) printf (nenter the value to second matrix) input (brc) multiply (abcrcc) printf (n The resultant Matrix is) display (crc) getch ( ) Passing string to functions The string are treated as character array in C and therefore the rules of passing string to function are very similar to those for passing array to functions
- WRITE ( ) (A (I) I) PRACTICE Write a program to take input to one dimensional array and display
- INTEGER IN ( ) PRINT ENTER THE ELEMENTS TO ARRAY READ ( ) (IN (I) I) PRINT THE ELEMENTS OF ARRAY IS WRITE ( ) (IN (I) I) STOP END Write a program to arrange one dimensional array in ascending order in FORTRAN
- PRACTICE Write a program to take input and display two dimensional arrays
- C input and display elements OF MATRIX INTEGER IN ( ) PRINT ENTER THE ELEMENTS TO ARRAY READ ( ) ((IN (IJ) J) I) PRINT THE ELEMENTS OF ARRAY IS WRITE ( ) ((IN (IJ) J) I) STOP END Write a program to add two matrixes and display the resultant matrix in FORTRAN
- C ADDING CORRESPONDING ELEMENT OF TWO MATRIX TO THIRD MATRIX INTEGER A ( ) B ( ) C ( ) RRCC PRINT ENTER THE ROW AND COLUMN OF FIRST MATRIX READ RC

PRINT ENTER THE ROW AND COLUMN OF SECOND MATRIX READ RC IF ((REQR) AND - (CEQC)) THEN PRINT ENTER ELEMENT TO FIRST ARRAY DO IR DO JC READ A (IJ) CONTINUE PRINT ENTER ELEMENT TO SECOND ARRAY DO IR DO JC READ B (IJ) CONTINUE DO IR DO JC C (IJ) A (IJ) B (IJ) CONTINUE PRINT RESULTANT ARRAY IS DO IR DO JC PRINT C (IJ) CONTINUE ELSE PRINT ADDITION IS NOT POSSIBLE ENDIF STOP END Write a program perform matrix multiplication using nested DO loop C MATRIX MULTIPICATION INTEGER A ( ) B ( ) C ( ) RRCC PRINT ENTER THE ROW AND COLUMN OF FIRST MATRIX READ RC PRINT ENTER THE ROW AND COLUMN OF SECOND MATRIX READ RC IF (REQC) THEN PRINT ENTER ELEMENT TO FIRST ARRAY DO IR DO JC READ A (IJ) CONTINUE PRINT ENTER ELEMENT TO SECOND ARRAY DO IR DO JC READ B (IJ) CONTINUE DO IR DO JC C (IJ) DO KR C (IJ) C (IJ) A (IK) B (KJ) CONTINUE PRINT RESULTANT ARRAY IS DO IR DO JC PRINT C (IJ) CONTINUE ELSE PRINT MULTIPICATION IS NOT POSSIBLE ENDIF STOP END Write a program to perform matrix multiplication using implied DO loop C MATRIX MULTIPICATION INTEGER A ( ) B ( ) C ( ) RRCC PRINT ENTER THE ROW AND COLUMN OF FIRST MATRIX READ RC PRINT ENTER THE ROW AND COLUMN OF SECOND MATRIX READ RC IF (REQC) THEN PRINT ENTER ELEMENT TO FIRST ARRAY READ ( ) ((A (IJ) J C) IR) PRINT ENTER ELEMENT TO SECOND ARRAY READ ( ) ((B (IJ) JC) IR) DO IR DO JC C (IJ) DO KC C (IJ) C (IJ) A (IK) B (KJ) CONTINUE PRINT RESULTANT ARRAY IS WRITE ( ) ((C (IJ) JC) IR) ELSE PRINT MULTIPICATION IS NOT POSSIBLE ENDIF STOP END Write a program to sum the following series up to n term xxx

- ++++

- indexn For example int a We can write the following statement a x a printf (d a) printf (d a) If we want to read all the elements of above array we can make a loop like for (i i include void main ( ) clrscr ( ) for (i i include main ( ) int a b s i j clrscr ( ) printf (Enter first matrix n) for (i i include main ( ) int a b s int m n l p i j k clrscr ( ) printf (Enter row of first matrix ( ) t) scanf (d m) printf (Enter column of first matrix ( ) t) scanf (d n) printf (Enter row of second matrix ( ) t) printf (Enter column of second matrix ( ) t) scanf (d p) if (nl) printf (Multiplication is not possible) else printf (Enter the first matrix n) for (i im i) for (j jn j) printf (Enter ad d t i j) scanf (d a i j printf (Enter the second matrix n) for (i il i) for (j jp j) printf (Enter bd d t i j) scanf (d bi j) for (i im i) for (j jp j) si j for (i im i) for (j jp j) for (k kn k) si j si j ai k bk j printf (The matrix multiplication is n) for (i im i) for (j jp j) printf (dt si j printf (n) getch ( ) end of main ( ) Output Enter row of the first matrix ( ) Enter column of the first matix ( ) Enter row of the second matrix ( ) Enter column of the second matrix ( ) Enter the first matrix Enter a

Enter a Enter the second matrix Enter b Enter b The matrix multiplication is Solve the Matrix b ( ) a s d c b a B h g f e A dh cg dg ce bh af bg ae) AB Passing arrays to functions Like any other variables we can also pass entire array to a function

- Program to illustrate passing array to function include void display (int) function prototype main ( ) int num i clrscr ( ) printf (n The content of array is n) for (i i include int sum (int a) void output (int a) main ( ) int a s i clrscr ( ) printf (Enter elements n) for (i i include void input (int t) void output (int t) int t) main ( ) int a b clrscr ( ) printf (Enter first array n) printf (Enter second array n) output (ab) getch ( ) void input (int t) int ij for (i i i) for (j j j) scanf (d t i j) void output (int t int t) int ij int s for (i i i) for (j j j) si j ti j ti j for (i i i) for ( ) j j) printf (dt si j) printf (n) Output Enter first array Enter second array Subtraction is Arrays of Strings String Array of character type which is terminated by null characters is known as string

- Array name is pointer to itself

- We know that array name is also pointer to so the above program can be written as include include main ( ) int a int i clrscr ( ) for (i i i) printf (dn (ai) getch ( ) output Pointer and Multidimensional Array We know that the name of array is point to first element of array ie

merged_topic_18: pointer, operator, subtract, integers, operators, precedence, associativity, left, right

- C operators can be classified into a number of categories

- Special operators C support some special operators of interest such as comma operator sizeof operator pointer operator (and) and member selection operator (

- Example X (int) ie is converted to integer by truncation Operator precedence and associativity Precedence is used to determine how an expression involving more than one operator is evaluated

- Example int a int pa Pointer Expression C allows us to add integers to or subtract integers from pointers as well as to subtract one pointer from another

- Pppp In addition to arithmetic operations discussed pointer can also be compared using the relational operators

- We may no use pointer in division or multiplication

- Pointer increment and scalar factor When we increment a pointer its value is increased by the length of the data type that it points to

- ++++
- Precedence Associativity Precedence of an operator in C is defined as the order of evaluation
- The operators are grouped hierarchically according to their precedence
- Operations with a high precedence are carried out before operation having a lower precedence
- Within each of the precedence groups described above the associativity is left to right
- Operators not equal to These operators all fall within the same precedence group which is lower than the arithmetic and unary operators
- The associativity of these operators is left to right
- The associativity of size of right to left
- Give a table that shows their precedence and associatively
- The dot operator has precedence and associativity is left to right

merged_topic_19: satisfied, loop, body, looping, condition, stopping, terminate, problem, algorithm

- Decision making and looping Looping (or iteration) is the process of executing a sequence of statements until some condition for termination of loop is satisfied
- If the conditions are not satisfied then the body of the loop will not be executed
- If the condition is true the program continues to evaluate the body of the loop once again
- If the condition is true the body of loop is executed otherwise the loop is terminated and execution continues with the statement that immediately follows the loop
- If the condition is satisfied the body of the loop is again executed
- ++++
- After performing the instructions that is after the algorithm terminates the desired results must be obtained
- Step Stop (End) The above mentioned example is simpler one but development of an algorithm of a complex problem is very difficult
- Sometimes it becomes necessary to come out of the loop even before the loop condition becomes false
- In such a situation break statement is used to terminate the loop
- This statement causes an immediate exit from that loop in which this statement

appears
- The loop does not terminate when a continue statement is encountered
- Jumping completely out of the loop under certain conditions terminating the execution of a loop
- Problem statement must include a stopping condition ie
- The problem can be solved in this manner for any value of n greater than (n represents a stopping condition)

merged_topic_20: function, calling, return, does, receive, recursive, calls, temperature, satisfied, recursion

- Void data type (for empty set of value and nonreturning functions) the void type specifies an empty set of values
- It is used as the return type for functions that do not return a value
- Function with no arguments and no return values When a function has no argument it does not receive any data from the calling function
- Similarly when it does not return value the calling function does not receive any data from the called function
- Function with no argument and return type In this function the function does not take any input from the calling function but it returns value to the calling function
- Recursive A function is said to be recursive if a statement in the body of the function calls itself
- The return statement is necessary only when the function is returning some data back to the calling function
- ++++
- For example inlcude main ( ) int c double d crlscr ( ) printf (Enter temperature in Celsius) scanf (d c) d convert (c) Function call printf (The Fahrenheit temperature of d C lf Fc d) getch ( ) double convert (int C) function definition double f f c return f What is about main ( ) function
- main ( ) register int a Recursive Function Recursion in programming is a technique for defining a problem in terms of one or more smaller versions of the same problem
- A recursive function is one that calls itself directly or indirectly to solve a smaller version of its task until a final call which does not require a selfcall
- Thus the function is called recursive function if it calls to itself and recursion is a

process by which a function call itself repeatedly until some specified condition will be satisfied

- To solve a problem using recursive method two conditions must be satisfied
- we must have an if statement somewhere to force the function to return without the recursive call being executed otherwise function will never return
- What is a recursive function

merged_topic_21: prototype, declaration, local, function, global, definition, body, display, int

- Function declaration or Function prototype
- Function Prototype A function prototype is a declaration of the function that tells the program about the type of the value returned by the function and the number and type of each argument
- Functiontype functionname (parameter list) Example void sum (int a int b) when we place the declaration above all the function (in the global declaration section) the prototype is referred to as global prototype
- When we place it in a function definition (in the local declaration section) the prototype is called a local prototype
- The prototype declaration should be similar to the function header
- ++++
- For example int add (int int) int add (int a int b) void display (int a) void display (int) Function definition A function definition is a group of statements that is executed when it is called from some point of the program
- statements is the functions body
- The first line of the function definition is known as function header
- This statement can appear anywhere inside the body of the function
- Formal Actual When a function is called some parameters are written within parenthesis
- What is a function
- When we call the function no need to subscript or square brackets

merged_topic_22: function, calling, return, does, receive, parameter, type, returned, mul

- Void data type (for empty set of value and nonreturning functions) the void type specifies an empty set of values
- It is used as the return type for functions that do not return a value
- Function with no arguments and no return values When a function has no argument it does not receive any data from the calling function
- Similarly when it does not return value the calling function does not receive any data from the called function
- Function with no argument and return type In this function the function does not take any input from the calling function but it returns value to the calling function
- Recursive A function is said to be recursive if a statement in the body of the function calls itself
- The return statement is necessary only when the function is returning some data back to the calling function
- ++++
- The general syntax of function prototype is return_type function_name (type type typen) where return_type specifies the data type of the value returned by the function
- A function can return value of any data type
- The general syntax is return_type function_name (parameter parameter parametern) statements where return_type is the data type specifier of data returned by the function
- The syntax of teh call if function return_type is void is function_name (parameter name) If function return int float or any other type value then we have to assign the call to same type value like variable function_name (parameter) For example m mul (x y) The type of m is same as the return type of mul function
- This type of function is defined as void function_name ( ) body of function The keyword void means the function does not return any value

merged_topic_23: function, calling, return, does, receive, sum, arguments, func

- Void data type (for empty set of value and nonreturning functions) the void type specifies an empty set of values
- It is used as the return type for functions that do not return a value
- Function with no arguments and no return values When a function has no argument it

does not receive any data from the calling function

- Similarly when it does not return value the calling function does not receive any data from the called function

- Function with no argument and return type In this function the function does not take any input from the calling function but it returns value to the calling function

- Recursive A function is said to be recursive if a statement in the body of the function calls itself

- The return statement is necessary only when the function is returning some data back to the calling function

- ++++

- int addition (int a int b) function prototype main ( ) int addition ( ) function call printf (The result isdz) int addition (int a int b) function header int r r ab return r Output The result is return statement The return statement is used in a function to return a value to the calling function

- Program to understand the use of return statement include void funct (int float) main ( ) int age float ht clrscr ( ) printf (Enter age and height) scanf (df age ht) funct (age ht) getch ( ) void funct (int age float ht) if (age) printf (Age should be less than n) return if (ht) printf (Height should be more than n return print (selected n) The second form of return statement is used to terminate a function and return a value to the calling function

- For example return return (xyz) return (sum (ab)) Calling a function (or A calling function) A function can be called by specifying its name followed by a list of arguments enclosed in parenthesis and separated by commas in the main ( ) function

- Functions with no arguments and no return value

- Functions with no arguments and a return value functions with arguments and no return value Functions with no arguments and no return value When a function has no arguments it does not receive any data from the calling function

- Program to illustrate the function with no arguments and no return values void add ( ) int a b sum printf (n Enter two numbers t) scanf (dd a b) sum a b printf (n The sum is d sum) void main ( ) clrscr ( ) add ( ) getch ( ) Output Enter two numbers The sum is Functions with no arguments and a return value These type of functions do not receive any arguments but they can return a value

- int func (void) main ( ) int r int func (void) return (expressions) Program to illustrate function with no argument and a return value inlcude int add (void) main) printf (The sum is dn add ( )) int add (void) int n n sum printf (Enter two numbers t)

scanf (dd n n) sum nn return (sum) Output Enter two numbers The sum is Functions with arguments and no return values These types of functions have arguments hence the calling function can send data to the called function but the called function does not return any value

- These functions can be written as void func (int int) main ( ) func (a b) void func (int c int d) statements Program to illustrate the functions with arguments but no return values void add (int int) main ( ) int a b printf (Enter two numbers t) scanf (dd a b) add (a b) void add (int c int d) int sum sum cd printf (n The sum is d sum) Output Enter two numbers The sum is Functions with arguments and return values These types of functions have arguments so the calling function can send data to the called function it can also return any values to the calling function using return statement
- This function can be written as int func (int int) main ( ) int r r func (ab) int func (int a int b) return (expression) Program to illustrate the function with argument and return values int add (int int) main ( ) int a b sum clrscr ( ) printf (Enter two numbers t) scanf (dd a b) sum add (a b) printf (ln the sum is td sum) int add (int a int b) int sum sum ab return sum Output Enter two numbers The sum is Call by value Call by reference (Or Pass by value Pass by reference) Passing arguments to functions An argument is a data passed from a program to the function

merged_topic_24: pass, calling, called, function, parameters, reference, passed, mechanism, changed

- In effect there is no data transfer between the calling function and the called function
- It is a twoway data communication between the calling and called function
- Ways of passing arguments to functions There are two different mechanisms to pass arguments to function
- Pass by value (also called as call by value) Pass by reference (also known as call by pointer) Pass by value In pass by value values of actual parameters are copied to the variable in the parameter list of the called function
- The called function works on the copy and not on the original values of the actual parameters
- This ensures that the original data in the calling function can not be changed accidently
- In this case the called function directly work on the data in the calling function and the

changed value are available in the calling function for its use
- This method is also used when we require multiple values to returned by the called function
- Let us consider following example to explain it
- ++++
- There are some variables that are used in more than one function
- When some specific code is to be used more than once and at different places in the program the use of function avoids repetition of that code
- In function we can pass a variable by two ways
- Pass by value (or call by value)
- Pass by reference (or call by reference) Function call by value In this the value of actual parameter is passed to formal parameter when we call the function
- its value persists between different function calls
- When we pass array that pass as a call by reference because the array name is address for that array
- As address is passed in this case this mechanism is also ka call by address or call by reference
- As address of variable is passed in this mechanism if value in the passed address is changed within function the value of actual variable also changed

merged_topic_25: pointer, tells, slide, asterisk, unrelated, address, price, equivalent, variable

- Pointer refer to book for proper knowledge as this content is from class note slide Declaring Pointer Variables Syntax data_type pt_name This tells the compiler three things about the variable pt_name
- This asterisk ( ) tells that the variable pt_name is a pointer variable
- pt_name needs a memory location
- pt_name points to a variable type of data_type
- However any comparisons of pointer that refers to separate and unrelated variables make no sense
- Similarly two pointer can not be added
- ++++
- Thus pointer variables are defined as int a float b char c where a b c are pointer

variable which stores address of integer float and char variable

- Program to print address of variable using include main ( ) int sn float price printf (value of snd Address of snun sn sn) printf (value of pricef Address of priceun price price) output value of sn Address of sn value of price Address of price Pointer Fundament Pointer Declaration Introduction A pointer is a variable that contains a memory address of data or another variable

- The asterisk preceding this name informs the compiler that the variable is declared as a int iptrsn float fptrprice iptrsn assigning of pointer fptrprice Now iptr contains the address of variable sn ie

- it points to variable sn similarly fptr points to variable sn similarly fptr points variable price

- If pointers are declare after the variable like int snpsn float priceqprice It is also possible to assign the value of one pointer variable to the other provided their base type is some

- PP Now both pointer variable P and P contains the address of variable sn and points the same variable iptr sn fptr P sn P We can also assign constant zero to a pointer of any type

- Let us see some other examples P is equivalent to a (P) is equivalent to a xP is equivalent to xb printf (dfPP) is equivalent to printf (dfab) scanf (dfPP) is equivalent to scanf (dfab) include main ( ) int a float b int Pa printf (value of P address of a unP) printf (value of P address of b un P) printf (Address of P unP) printf (Address of P un P) printf (value of a dddnaP (a)) printf (value of b fffnbP (b) OUTPUT Value of P Address of a Value of P Address of b Address of P Address of P Value of a Value of b Passing pointers to functions A pointer can be passed to a function as an argument

- What does p p and p represents if p is declared as integer pointer

merged_topic_26: file, opening, data, defined, secondary, mode, opened, fopen, open

- The basic file operations performed are Naming a file Opening a file Reading data from a file Writing data from a file Closing a file

- Defining and opening a file If we want to perform operation in a file in the secondary memory we must specify certain things about the file to the operating system

- Example Studenttxt Employdat Data structure of a file is defined as FILE in the library

of standard IO function definitions

- Therefore all files should be declared as type FILE before they are used
- FILE is a defined data type
- For example we may write data to the file or read the already existing data
- Following is the general format for declaring and opening a file FILE fp fp fopen (filenamemode) the first statement declares the variable fp as a pointer to the data type FILE
- The second statement opens the file named filename and assigns an identifier to the FILE type pointer fp
- ++++
- fp fopen (argv w) opens a file with name TEXT
- The function fopen ( ) is used to open a file
- On success fopen ( ) returns a pointer of type FILE and on error it returns NULL
- For example FILE fp fp fp fopen (myfiletxt w) fp fopen (yourfiledat r) The file opening mode specifies the way in which a file should be opened (ie
- To open a file in binary mode we can append b to the mode and to open the file in text mode t can be appended to the mode
- But since text mode is the default mode t is generally omitted while opening files in text mode
- For example wb Binary file opened in write mode ab or (ab) Binary file opened in append mode rt or (rt) Text file opened in update mode w or (wt) Text file opened in write mode The file that was opened using fopen ( ) function must be closed when no more operations are to be performed on it
- Explain in brief the various file opening modes

merged_topic_27: columns, rows, varies, array, number, dimensional, multidimensional, arrays, square

- A two dimensional array is the simplest of multidimensional arrays
- For instance an array A MN is a M by N table with M rows and N columns containing M X N element
- The number of elements in a D array can be determined by multiplying number of rows with number or columns
- For example the number of elements in an array A is calculated as X

- N AN Row M A The general form of a twodimensional array is type arraynamerow_sizecolumn_size Where type is the data type of the array having name arrayname row_size the first index refers to the number of rows in the array and column_size the second index refers to the number of columns in the array
- For example consider a two dimensional array A (IJ) where I varies from to and J varies from to
- ++++
- Square brackets enclosing the size specification for text
- auto register static or extern) Single and Multidimension arrays One or single dimensional array There are several forms of an array in Cone dimensional and multidimensional array
- In one dimensional array there is a single subscript or index whose value refers to the individual array element which ranges form to n where n is the size of the array
- Examples of single dimensional array Program that reads integers from keyboard and displays entered numbers in the screen include main ( ) int a i clrscr ( ) printf (Enter numbers t) for (i i main ( ) float a int i clrscr ( ) printf (The continuous memory locations are n) printf (tu ai) address of array element getch ( ) Output The continuous memory locations are Program to sort n numbers in ascending order main ( ) int num i j n temp clrscr ( ) printf (How many numbers are there
- Enter numbers The numbers in ascending order Multidimensional arrays An array of arrays is called multidimensional array
- For example a one dimensional array of one dimensional array is called two dimensional array
- A one dimensional array of two dimensional arrays is called three dimensional arrays etc
- Declaration of two dimensional array Multidimensional arrays are declared in the same manner as one dimensional array except that a separate pair of square brackets are required for each subscript ie
- two dimensional array requires two pair of square bracket three dimensional array requires three pairs of square brackets four dimensional array require four pair of square brackets etc
- For example int n float a static char line double add Initialization of multidimensional array Similar to one dimensional array multidimensional array can also be initialized

merged_topic_28: matrix, print, enter, column, array, dimensional,

multidimensional, arrays, square

- Store the sum of A and B in C
- Practice Write to program to arrange list of elements in ascending order include include include void main ( ) int ntemp int ij printf (nEnter the list of elements) for (iinj) tempni ninj njtemp printf (n The elements in assending order) for (iii) printf (ndni) getch ( ) Two Dimensional Array A twodimensional array is an array in which each element is itself an array
- Practice Write a program that adds the individual rows of a two dimensional array of m by n and store the sums of rows into a single dimensional array using functions
- Write a function that takes a two dimensional array and onedimensional array and process the result and store in onedimensional array
- Matrix Multiplication include include void input (int aint rint c) int ij for (iiri) for (jjcj) scanf (daij) void display (int aint rint c) int ij for (iiri) for (jjcj) printf (daij) printf (n) void multiply (int aint bint cint rint cint c) int ijk for (iiri) for (jjcj) cij for (kkck) cijaikbkj void main ( ) int abcrrcc clrscr ( ) printf (nEnter the row and column for first matrix) scanf (ddrc) printf (nEnter the row and column for second matrix) scanf (ddrc) if (cr) printf (nMultipication is not possible) else printf (nenter the value to first matrix) input (arc) printf (nenter the value to second matrix) input (brc) multiply (abcrcc) printf (n The resultant Matrix is) display (crc) getch ( ) Passing string to functions The string are treated as character array in C and therefore the rules of passing string to function are very similar to those for passing array to functions
- WRITE ( ) (A (I) I) PRACTICE Write a program to take input to one dimensional array and display
- INTEGER IN ( ) PRINT ENTER THE ELEMENTS TO ARRAY READ ( ) (IN (I) I) PRINT THE ELEMENTS OF ARRAY IS WRITE ( ) (IN (I) I) STOP END Write a program to arrange one dimensional array in ascending order in FORTRAN
- PRACTICE Write a program to take input and display two dimensional arrays
- C input and display elements OF MATRIX INTEGER IN ( ) PRINT ENTER THE ELEMENTS TO ARRAY READ ( ) ((IN (IJ) J) I) PRINT THE ELEMENTS OF ARRAY IS WRITE ( ) ((IN (IJ) J) I) STOP END Write a program to add two matrixes and display the resultant matrix in FORTRAN
- C ADDING CORRESPONDING ELEMENT OF TWO MATRIX TO THIRD MATRIX INTEGER A ( ) B ( ) C ( ) RRCC PRINT ENTER THE ROW AND COLUMN OF FIRST MATRIX READ RC

PRINT ENTER THE ROW AND COLUMN OF SECOND MATRIX READ RC IF ((REQR) AND - (CEQC)) THEN PRINT ENTER ELEMENT TO FIRST ARRAY DO IR DO JC READ A (IJ) CONTINUE PRINT ENTER ELEMENT TO SECOND ARRAY DO IR DO JC READ B (IJ) CONTINUE DO IR DO JC C (IJ) A (IJ) B (IJ) CONTINUE PRINT RESULTANT ARRAY IS DO IR DO JC PRINT C (IJ) CONTINUE ELSE PRINT ADDITION IS NOT POSSIBLE ENDIF STOP END Write a program perform matrix multiplication using nested DO loop C MATRIX MULTIPICATION INTEGER A ( ) B ( ) C ( ) RRCC PRINT ENTER THE ROW AND COLUMN OF FIRST MATRIX READ RC PRINT ENTER THE ROW AND COLUMN OF SECOND MATRIX READ RC IF (REQC) THEN PRINT ENTER ELEMENT TO FIRST ARRAY DO IR DO JC READ A (IJ) CONTINUE PRINT ENTER ELEMENT TO SECOND ARRAY DO IR DO JC READ B (IJ) CONTINUE DO IR DO JC C (IJ) DO KR C (IJ) C (IJ) A (IK) B (KJ) CONTINUE PRINT RESULTANT ARRAY IS DO IR DO JC PRINT C (IJ) CONTINUE ELSE PRINT MULTIPICATION IS NOT POSSIBLE ENDIF STOP END Write a program to perform matrix multiplication using implied DO loop C MATRIX MULTIPICATION INTEGER A ( ) B ( ) C ( ) RRCC PRINT ENTER THE ROW AND COLUMN OF FIRST MATRIX READ RC PRINT ENTER THE ROW AND COLUMN OF SECOND MATRIX READ RC IF (REQC) THEN PRINT ENTER ELEMENT TO FIRST ARRAY READ ( ) ((A (IJ) J C) IR) PRINT ENTER ELEMENT TO SECOND ARRAY READ ( ) ((B (IJ) JC) IR) DO IR DO JC C (IJ) DO KC C (IJ) C (IJ) A (IK) B (KJ) CONTINUE PRINT RESULTANT ARRAY IS WRITE ( ) ((C (IJ) JC) IR) ELSE PRINT MULTIPICATION IS NOT POSSIBLE ENDIF STOP END Write a program to sum the following series up to n term xxx

- ++++

- Square brackets enclosing the size specification for text

- auto register static or extern) Single and Multidimension arrays One or single dimensional array There are several forms of an array in Cone dimensional and multidimensional array

- In one dimensional array there is a single subscript or index whose value refers to the individual array element which ranges form to n where n is the size of the array

- Examples of single dimensional array Program that reads integers from keyboard and displays entered numbers in the screen include main ( ) int a i clrscr ( ) printf (Enter numbers t) for (i i main ( ) float a int i clrscr ( ) printf (The continuous memory locations are n) printf (tu ai) address of array element getch ( ) Output The continuous memory locations are Program to sort n numbers in ascending order main ( ) int num i j n temp clrscr ( ) printf (How many numbers are there

- Enter numbers The numbers in ascending order Multidimensional arrays An array of

arrays is called multidimensional array

- For example a one dimensional array of one dimensional array is called two dimensional array

- A one dimensional array of two dimensional arrays is called three dimensional arrays etc

- Declaration of two dimensional array Multidimensional arrays are declared in the same manner as one dimensional array except that a separate pair of square brackets are required for each subscript ie

- two dimensional array requires two pair of square bracket three dimensional array requires three pairs of square brackets four dimensional array require four pair of square brackets etc

- For example int n float a static char line double add Initialization of multidimensional array Similar to one dimensional array multidimensional array can also be initialized

merged_topic_29: index, base, address, arrayindex, noriginal, local, swap, prints, stored

- Identifiers refer to the name of variables functions and arrays

- In c arrays index numbering starts with

- Example include include void main ( ) int xbx int yay int z clrscr ( ) printf (nEnter the value) scanf (dx) printf (nEnter the value) scanf (dy) zxyab printf (nThe value of z isdz) printf (nOriginal value of b is ub) b printf (nChanged value of b is ub) printf (nOriginal value of a is ua) aa printf (nChanged value of a is ua) getch ( ) Pointer and Arrays When an array is declared the compiler allocate a base address and sufficient amount of storage to contain all the elements of the array in contiguous memory allocation

- The base address is the location of the first element (index) of the array

- The compiler also defines the array name as a constant pointer to the first element

- Suppose we declare an array x as follows int x Suppose the base address of x is then Element x x x x x Value Address Example Heres the array version include include void main ( ) int a int i clrscr ( ) for (iii) printf (nd uaiai) getch ( ) Now lets see how this program would look using pointer notation

- (array index) is the same as arrayindex

- (array index) is the same as arrayindex Pointer to Array in Function As an example to

explain that a program where each element of array will be added by a constant

- ++++

- int x float y char c a x the address of x is assigned to pointer variable a b y the address of y is stored to pointer variable b c c the address of c is stored to pointer variable c include void swap (int int) function prototype main ( ) int x y x clrscr ( ) swap (x y) function call by address printf (xdn x) printf (ydn y) getch ( ) void swap (int a int b) int t t a a b b t Output x y PU Concept of Local Global Static variables Local variables (automatic or internal variable) The variables that are defined within the body of a function or a block are local to that function or block only and are called local variables

- Address operator ( ) C provides an address operator which returns the address of a variable when placed before it

- The following program prints the address of variabvles using address operator

- Let us take an example int a float b intP a floatp b In above program if we place before P when we can access the variable whose address is stored in P

- For example int aP Pa After statement pa p point the array ie

- the p contains the address of a (first address of array) P After this statement P points to a element

- When i then (pi) is (p) means p therefore this prints the value which store at location P Similarly (p) means the value of second element because p is pointer to the second element of array

- p contains the address of element) or p is name of array

merged_topic_30: loop, exitcontrolled, tested, entrycontrolled, control, body, executed, condition, statements

- EntryControlled loop In entrycontrolled loop the control conditions are tested before the start of the loop execution

- ExitControlled loop In exitcontrolled loop the test is performed at the end of the body of the loop and therefore the body is executed unconditionally for the first time

- When the body of the loop is executed the control is transferred back to the for statement after the evaluating the last statement in the loop

- Now the control variable is incremented using as assignment statement such as ii and the new value of the control variable is again tested to see whether it satisfies the loop

condition

- ++++

- This ensures that the algorithm will ultimately terminate

- So a loop may be defined as a block of statements which are repeatedly executed for a certain number of times or until a particular condition is satisfied

- The control statement in loop decides whether the body is to be executed or not

- After the execution again the condition is checked and if it is found to be true then again the statements in the body of loop are executed

- The body of this loop may contain a single statement or a block of statements

- If the condition is true then again the loop body is executed and this process continues until the condition becomes false

- Instead the remaining loop statements are skipped and the computation proceeds directly to the next pass through the loop

merged_topic_31: statement, condition, expression, switch, statements, case, choice, default

- respectively Decision making and Branching C language possesses such decisionmaking capabilities by supporting the following statements if statement switch statement conditional operator statement goto statements These statements are known as decisionmaking statements

- if statement It takes the following form if (test expression) It allows the computer to evaluate the expression first and then depending on whether the value of the expression (relation or condition) is true (non zero) or false (zero) it transfers the control to a particular statement

- This point of program has two paths to follow one for the true condition and the other for the false condition Entry False True The ifelse statement The general form is if (test expression) Trueblock statement (s) else Falseblock statement (s) statementx if the test expression is true then the trueblock statement (s) immediately following the if statements are executed otherwise the falseblock statement (s) are executed

- In either case trueblock or falseblock will be executed not both

- Trueblock statement Falseblock statement Statement x Nesting of ifelse statement The general form is if (test condition) if (test condition) statement else statement else statement statementx The else if ladder A multipath decision is a chain of ifs in

which the statement associated with each else is an if

- It takes the following general form if (condition) statement else if (condition) statement else if (condition) statement else if (condition n) statementn else defaultstatement statementx The switch statement The switch statement tests the value f a given variable (or expression) against a list of case values and when a match is found a block of statements associated with that case is executed

- The general form of the switch statement is as shown below switch (expression) case value block break case value block break default defaultblock break statementx The expression is an integer expression or characters

- block blockare statement lists and may contain zero or more statements

- The general form is If (expression) nnn Where (expression) is a valid FORTRAN arithmetic expression enclosed within parenthesis

- The general form of the statement is IF (condn) Statement Where condn is a logical condition statement is an executable statement

- The general form of the statement is IF (condn) THEN S S ELSE S S ENDIF Where condn is a logical condition SS are the statement to be executed when cond is true SS are the statement to be executed when cond is false

- ++++

- For making this choice we use the switch statement

- The general syntax is switch (expression) case constant statements break case constantN statements break default statements Here switch case and default are keywords

- The expression following the switch keyword can be any C expression that yields an integer value or a character value

- Writing a switch statement inside another is called nesting of switches

- Firstly the switch expression is evaluated then value of this expression is compared one by one with every case constant

- If none of the case constant matches with the value of the expression then the block of statements under default is Flowchart switch (expression) case constant (expression constant) Body of case constant case constant (expression constant) Body of case constant case N default default Out of switch fig Flowchart of switch statement Program to understand the switch control statment include main ( ) int choice clrscr ( ) printf (Enter your choice) scanf (d choice) switch (choice) case case printf (secondn) case printf (thirdn) default printf (wrong choicen) getch ( ) Output Enter your choice Second Third Wrong Choice Here value of choice matches with second case so all the

statements after case are executed sequentially

- Write a menu driven program using switch statement having the following options

merged_topic_32: starting, variable, integer, letters, ijklm, root, declarations, contains, short, long

- Variable definitions The statement datatype variable_name eg
- int atotal Defining a variable tells the compiler the name of the variable and the type of variable
- An identifier declared as int becomes an integer variable and can hold integer value only
- An identifier declared as float becomes a floatingpoint variable and can hold floatingpoint number only
- This is done as follows register auto int a Most compilers allow only int or char type variable to be placed in register
- If the variables are not declared the variable names starting with letters IJKLM or N are considered to be integer variable and other as real variables
- The general format for declaring integer variables is shown below INTEGER list of variables Eg
- Example IMPLICIT INTEGER (A) The above statement declares that all the variable name starting with the alphabet A are integer variables
- This does not affect the general rule that the variables starting with the letters IJKLM or N are also integer variable
- So the above statement declare that all the variable starting with the letters AIJKLM or N are integer variable in the program
- ++++
- The standard keywords are auto break case char const continue default do double else enum extern float for goto if int long register return short signed sizeof static struct switch typedef union unsigned void volatile while The keywords are all lowercase
- A C program contains the following lines char d a b c a b d a a b c a b d w The first two lines are not type declaration which state that a b and c are integer variables and that d is a character type
- A C program contains the following type declarations int a b c float root root char flag text Thus a b and c are declared to be integer variables root and root are floating

variables flag is a chartype variable and text is an element chartype array
- These declarations could also have been written as follows int a int b int c float root float root char flag char text A C program contains the following type declarations short int a b c long int r s t int p q Also written as short a b c long r s t int p q short and short int are equivalent as are long and long int
- A C program contains the following type declarations float c c c double root root also written as long float root root A C program contains the following type declarations
- int c char star float sum Thus c is an integer variable whose initial value is star is a char type variable initially assigned the character sum is a floating point variable whose initial value is and factor is double precision variable whose initial value is
- A C program contains the following type declarations

merged_topic_33: pointer, operator, subtract, integers, operators, operand, operands, requires

- C operators can be classified into a number of categories
- Special operators C support some special operators of interest such as comma operator sizeof operator pointer operator (and) and member selection operator (
- Example X (int) ie is converted to integer by truncation Operator precedence and associativity Precedence is used to determine how an expression involving more than one operator is evaluated
- Example int a int pa Pointer Expression C allows us to add integers to or subtract integers from pointers as well as to subtract one pointer from another
- Pppp In addition to arithmetic operations discussed pointer can also be compared using the relational operators
- We may no use pointer in division or multiplication
- Pointer increment and scalar factor When we increment a pointer its value is increased by the length of the data type that it points to
- ++++
- Some operators require two operands while other act upon only one operand
- Most operators allow the individual operands to be expressions
- A few operators permit only single variable as operand
- The operands acted upon by arithmetic operators must represent numeric values
- The remainder operator ( ) requires that both operands be integers and the second

operand be non zero
- Similarly the division operator ( ) requires that the second operand be nonzero
- The operation always result in a truncated quotient (ie
- a x y delta sum ab area length width Assignment operator and equality operator are distinctly different
- These operators can not be used in place of one another
- The operand used with each of these operators must be a single variable

file1_topic_34: software, application, computer, disk, users

- Computer Software The software is what really gives the computer life
- Software is installed onto your computer and stored in mass storage devices
- Software can actually be placed into two separate categories system of applications
- System software System software (popularly called the operating system) is the foundation software for a computer
- An operating system (OS) controls all parts of the computer
- The major functions of the OS ate to handle all input devices (keyboard mouse disk) handle all output devices (screen printer disk) coordinate and manage use of other resources (memory disk CPU etc
-) accept commands from users provide an environment over which other programs (software) can run
- Examples of popular OS are DOS (Disk Operating System) Windows Unix Linux OS and Solaris etc
- Some of these OSes are specific to a particular computer system eg
- Solaris runs on Sun SPARC machines
- Application software Application software is the software that is developed for a particular use
- These software run over the environment provided by the system software
- There are thousands of application software
- There are other utility programs that check if the units inside your machine are functioning properly as they should
- Examples are the disk scanner that checks if the surface of your disks are damaged antivirus that checks if your computer has been infected with some virus (a malicious program that may just annoy you or even delete all the content of your hard drive) and clean if found etc

- Problem solving using Computers Standard software packages available in the market are intended for generalpurpose applications
- However the users often require customtailored software for performing specific data processing or computational tasks
- Application software development is the process of creating such software which satisfies the end users requirements and needs
- In simple language it can be said that problem solving using computers is the development of the application software
- Following steps needs to be followed for developing the application software

file1_topic_35: step, sum, goto, stop, series

- Spreadsheets very simply put gives you when you input
- For example Read numbers form user and display the resulting sum
- Write an algorithm and a flowchart to read a five number and check whether the number is a palindrome or not
- Algorithm Step Start Step Declare variable n r sum i Step i Step Repeat steps Step to until i is less than or equal to else goto step Step input to n Step sum Step tempn Step Repeat Steps to until n is greater than else goto step Step rn mod of Step sumsumr Step nn then goto Step Step if sum is equal to temp than goto Step else goto Step Step Display n as palindrome then goto step Step Display n is not palindrome Step ii then goto step Step Stop Flowchart False True True False True False Start Declare n r sum i i ii Is i
- Input to n Stop sum tempn Is n
- r n mod of sumsumr nn Is tempsum
- Display P li d Display Source code include include void main ( ) int nirsumtemp i clrscr ( ) while (i) rn sumsumr nn if (sumtemp) printf (Palindrome dtemp) else printf (Not Palindrome dtemp) i getch ( )
- Write a program to evaluate the following series using recursive function
- n include include float sum int sign void add (int nint i) int term if (in) signsign termiisign sumsumterm add (ni) else printf (n The sum of series is fsum) void main ( ) int n printf (nEnter the number of term) scanf (dn) add (n) getch ( ) Static variable A variable can be declared static using the keyword static like static int a Internal static variable are those which are declared inside a function
- PRACTICE XXX C SERIES USING ARITHMETIC IF INTEGER P PRINT ENTER THE VALUE OF

N AND X READ NX I P SUM IF (IN) SUMSUMXP PP II GOTO PRINT SUM IS SUM STOP END Logical IF statement The logical if condition checks any given logical condition and transfer the control accordingly

- Logical Condition The following are the relational operators and their symbols used in FORTRAN PRACTICE Write a program to display FIBNOACCI series until term value is less than in FORTRAN

- C FIBONACCI SERIES INTEGER FST PRINT ENTER THE NUMBER OF TERMS F S WRITE ( ) FS TFS IF (TGT) GOTO WRITE ( ) T FS ST GOTO STOP END IFTHENELSE statement The IFTHENELSE statement is more useful and easy to handle than the logical if statement

- ELSEIF PRACTICE Write a program to display Fibonacci series up to N term in FORTRAN

- C FIBONACCI SERIES INTEGER FST PRINT ENTER THE NUMBER OF TERMS READ ( ) N F S IF (NEQ) THEN WRITE ( ) F ELSE WRITE ( ) FS DO IN TFS WRITE ( ) T FS ST CONTINUE ENDIF STOP END Write a program to find the HCF for any two number entered by user in FORTRAN C WRITE A PROGRAM TO FIND HCF (GREATEST COMMON FACTOR) READ II IRIIII IF (IREQ) THEN PRINT THE HCF ISI STOP ENDIF II IIR GOTO STOP END Nested IFTHENELSE In certain cases we may have to use one IFTHENELSE structure within another IFTHEN

- ELSE ENDIF Write a program to check whether a given number is Armstrong or not in FORTRAN

- C PROGRAM TO FIND ARMSTRONG NUMBER OR NOT PRINT ENTER THE NUMBER READ N TEMPN SUM IF (NGT) THEN IRNN SUMSUMIR NN GOTO ENDIF IF (SUMEQTEMP) THEN PRINT TEMPIS ARMSTORNG ELSE PRINT TEMPIS NOT ARMSTRONG ENDIF STOP END DO LOOPS The DO LOOP is used whenever a particular job is to be repeated number of times

- If vv then stop otherwise go to step PRACTICE C PRIME NUMBER INTEGER COUNT READ NUM COUNT DO INUM IF (MOD (NUMI) EQ) THEN COUNTCOUNT ENDIF CONTINUE IF (COUNTEQ) THEN PRINT NUMIS PRIME ELSE PRINT NUMIS NOT PRIME ENDIF STOP END Rules for subscripted variables The following rules may be strictly followed while defining the subscripted variables

- PRINT ENTET THE VALUE OF N AND X READ NX P SIGN FACT SUM DO IN SIGNSIGN DEN DO JFACT DENDENJ CONTINUE NUMSIGNXP SUMSUMNUMDEN FACTFACT PP CONTINUE PRINT THE SUM ISSUM STOP END Additional Topic Look for Preprocessor and Dynamic Memory Allocation

file1_topic_36: large, console, volumes, oriented, data

- Databases are most useful for organizing data
- For example database software can keep track of your friends name address telephone numbers and you can add or remove whenever you want
- You need to be sure that the source of the data is consistent so that the date will be available in the future when you need it
- These are console oriented IO functions which always use the terminal (keyboard and screen) as the target place
- This works fine as long as the data is small
- However many real life problems involve large volumes of data and in such situations the console oriented IO operations pose two major problems
- It becomes cumbersome and time consuming to handle large volumes of data through terminals
- The entire data is lost when either the program is terminated or the computer is turned off
- It is therefore necessary to have more flexible approach where data can be stored on the disk and read whenever necessary without destroying the data
- This method employs the concept of files to store data

file1_topic_37: languages, level, high, lowlevel, programming

- Programming languages are mainly categorized into two types low level and high languages
- Lowlevel language A lowlevel language is a programming language much closer to the hardware
- High Level Language A high level language is a programming language that enables a programmer to write programs more or less independent of a particular type of computer
- Such languages are considered highlevel because they are closer to human languages and further from machine languages
- The main advantage of highlevel languages over lowlevel languages is that they are easier to read write and maintain
- The first highlevel programming languages were designed in the s
- low level or high level languages)

file1_topic_38: code, object, compiler, language, source

- This converting program is called assembler
- This can be done in two ways by a compiler or interpreter
- Compiler A compiler is a program that translates program (called source code) written in some high level language into object code
- The compiler derives its name from the way it works looking at the entire piece of source code and collecting and reorganizing the instructions
- A compiler translates highlevel instructions directly into machine language and this process is called compiling
- Compile Compiling is a process of transforming a program written in a highlevel programming language from source code into object code
- The first step is to pass the source code through a compiler which translates the highlevel language instructions in the source code follow its syntax rules
- The final step in producing an executable program after the compiler has produced object code is to pass the object code through a linker
- The linker combines molecules (different program segments) and gives real values to all symbolic addresses (memory locations) thereby producing machine code
- The compiler translates the source code into a form called object code
- Object code Object code is the code produced by a compiler
- Object code is often the same as or similar to a computers machine language
- The final step in producing an executable program is to transform the object code into machine languages if it is not already in this form
- A program called linker does this job
- Compilation and Execution Generally coding is done in highlevel language and sometimes in lowlevel language (such as assembly language)
- In a compiled language a translation program is run to convert the programmers entire highlevel program which is called the source code into a machine language code
- This translation process is called compilations
- The machine language code is called the object code and can be saved and either runs (executed) immediately or later
- In an interpreted language a translation program converts each program statement into machine code just before the program statement is to be executed
- No object code is stored and there is no compilation
- This means that in a program where one statement is executed several times (such as reading and employs payroll record) that statement is converted to machine language each time it is executed

- The compilation process The object of the compiler is to translate a program written in a high level programming language from source code to object code
- The first step is to pass the source code through a compiler which translate the high level language instructions into object code

file1_topic_39: allowed, illegal, pppp, expressions, rule

- This rule is called syntax
- The expressions such as pppp and pp are allowed
- pp or pp or p are not allowed
- ie pp Is illegal

file1_topic_40: languages, compiled, cobol, interpreted, compilers

- Now there are dozens of different languages such as Ada Algol BASIC COBOL C C FORTRAN LISP Pascal and Prolog etc
- Because compilers translate source code into object code which is unique for each type of computer many compilers are available for the same language
- For example there is a C compiler for PCs and another for Unix computers
- The high level languages can be either called compiled languages or interpreted languages
- Some of the most widely used compiled languages are COBOL C C FORTRAN etc
- Unlike the compiled languages
- The most frequently used interpreted language is BASIC

file1_topic_41: interpreter, faster, programs, interpreted, interpreters

- Interpreter An interpreter translates highlevel instructions into an intermediate form which it then executes
- Interpreter analyzes and executes each line of source code in succession without looking at the entire program the advantage of interpreters is that they can execute a program immediately
- However programs produced by compilers run much faster than the same programs executed by an interpreter

- Compiled programs generally run faster than interpreted programs
- The advantage of an interpreter however is that it does not need to get through the compilation stage during which machine instructions are generated
- The interpreter on the other hand can immediately execute highlevel programs
- For this reason interpreters are sometimes used during the development of a program when a programmer wants to add small sections at a time and test them quickly
- Compiler languages are better than interpreted languages as they can be executed faster and more efficiently once the object code has been obtained
- On the other hand interpreted languages do not need to create object code and so are usually easier to develop that is to code and test

file1_topic_42: step, flowchart, algorithm, lines, goto

- Algorithm Development and Flowcharting
- Algorithm Development and Flowchart (Program design) You know you have a problem and have identified it in the program analysis stage
- Algorithms Algorithms are a verbal or say written form of the program
- It can be defined as ordered description of instructions to be carried out in order to solve the given task
- Basic Guidelines for writing algorithms
- Describe everything clearly and explicitly
- Important Features of Algorithm
- Flowchart One of the most widely used devices for designing programs is the flowchart which graphically represents the logic needed to solve a programming problem
- A programming flowchart represents the detailed sequence of steps needed to solve the problem
- Program flowcharts are frequently used to visualize the logic and steps in processing
- In other words its a diagrammatic representation of algorithm
- Basic blocks used for drawing flowcharts Structure Purpose Start Stop Processing Decision making Input Outputs Connector For examples Read numbers form user and display the resulting sum
- Flowchart to find the largest among three entered numbers Y N Y N N Y start Read a b cab Display Stop Start input A B C Is AB Is AC Is BC Print A Print C Print B Stop Coding Writing the program is called coding

- Write an algorithm and flowchart to generate the following
- Algorithm Step start Step input to n Number of lines Step Initialize c as Deviation on each line Step i Number of lines Step Repeat step to until i less than or equal to n else goto step Step j Number of column Step Repeat step to until j less than or equal to (n) else goto step Step Is j greater than or equal to (nc) and less than or equal to (nc) than goto Step else goto Step Step Display then goto step Step Display Step jj then goto step Step Display newline Step cc Step ii then goto step Step Stop for (ii (nc) j include void main ( ) int ncij clrscr ( ) c deviation printf (nEnter the number of lines) scanf (dn) number of lines Flow Chart False True False True True False Start Input to n c ii (nc) j (nc)

file1_topic_43: requirements, specifying, inputs, processing, input

- Specifying the input requirements
- Specifying the processing requirements
- Specifying the input requirements Now that you have determined the outputs you need to define the input and data
- To do this you list the inputs required and the source of the data
- For example in a payroll program inputs could be employee timesheets and the source of the in put could be either the employees themselves or their supervisors
- Specifying the processing requirements Now you have to determine the processing requirements for converting the input data to output
- Quantities that have specified relation to inputs

file1_topic_44: positive, symbol, assumed, negative, sign

- Do not make any assumptions
- or will precede the signed numeric item
- (with o or x) causing octal and hex item to be preceded by and x
- Capital alphabetsAZ Digit Symbols
- The symbol or can occur only at the left most end of the number
- For negative number symbol is used at the most position and for the positive number symbol is used
- If no symbol occurs the number is assumed to be positive
- A negative number must be written with the symbol

- For a positive number the symbol is optional
- If there is no sign the number is assumed to be positive
- The exponent is always an integer with at most two digit
- The exponent can have sign (or)
- If there is no sign the exponent is assumed to be positive
- The subscript is always an integer
- The subscript value can not be negative

file1_topic_45: containing, implemented, makes, effectiveness, concise

- Effectiveness All operations must be sufficiently basic so as to be implemented with even paper and pencil
- In addition using many of them makes a program logic complicated and renders the program unreadable
- These enable us to develop concise programs containing repetitive processes without the use of goto statements
- That is they can not be used for reading a text containing more than one word

file1_topic_46: notation, scanf, field, reads, specified

- Display the value in C
- It can be performed using scanf function
- The general format of scanf is scanf (control stringargargarnn) The control string specifies the field format in which the data is to be entered and the arguments arglargargn specify the address of locations where the data is stored
- When the scanf reads a particular value reading of the value will be terminated as soon as the number of character specified by the field width is reached (if specified) or until a character that is not valid for the value being read in encountered
- For example scanf (d d dab) will assign the data as follows to a skipped (because of) to b Inputting Real Numbers Unlike integer numbers the field width of real number is not to be specified and therefore scanf reads real numbers using the simple specification f for both the notation namely decimal point notation and exponential notation

file1_topic_47: keyword, lowercase, colon, underscore, permitted

- Keyword All keyword have fixed meaning and these meaning can not be changed
- All keyword words must be written in lowercase
- Both uppercase and lowercase letter are permitted although lowercase letters are commonly used
- The underscore character is also permitted in identifiers
- First character must be an alphabet (or underscore)
- Can not use a keyword
- Note that case labels end with a colon ( ) The
- A label is any valid variable name and must be followed by a colon
- Note only single quote ( ) must be used and not the double quotes ( )
- No special is allowed in a variable name
- The subscript must be given within parenthesis after the variable name

file1_topic_48: digits, succeeding, alphabets, character, displays

- These are userdefined names and consist of a sequence of letters and digits with a letter as a first character
- This statement displays the character contained in the variable_name at the terminal
- Variable name can be form one to six character in length
- The first character of the variable name must be an alphabet and the succeeding characters can be alphabets or numeric digits

file1_topic_49: symbolic, names, verbs, fortran, naming

- The rules of identifiers
- Symbolic names have the same form as variable names
- A blank space is required between define and symbolic name and between the symbolic name and the constant
- After definition the symbolic name should not be assigned any other value within the program by using an assignment statement
- Symbolic names are NOT declared for data types
- Rules of naming variable
- FORTRAN verbs which have special meaning in FORTRAN can not be used as variable names

file1_topic_50: sizeof, const, size, returns, modified

- Example const int size const is a new data type qualifier defined by ANSI standard
- This tells the compiler that the value of the int variable size must not be modified by the program
- The sizeof operator The sizeof is a compile time operator and when used with an operand it returns the number of bytes the operator occupies
- sizeof (int) It returns
- The size must be integer value or integer constant without any sign

file1_topic_51: conversion, unsigned, signed, type, implicit

- In order to provide some control over the range of number and storage space c has three classes of integer storage namely short int int and long int in both signed and unsigned form
- Type Size (bits) Range char or signed char to unsigned char to int or signed int to unsigned int to short int or signed short int to unsigned short int to long int or signed long int to unsigned long int to float E to E double E to E long double E to E Operators Operators are words or symbols that cause a program to do something to variable
- Type conversion in expression Implicit type conversion C automatically converts any intermediate value to the proper type so that the expression can be evaluated without losing any significance
- This automatic conversion is known as implicit type conversion
- During evaluation it adheres to very strict rules of type conversion
- If the operands are of different type the lower type is automatically converted to the higher type before the operation proceeds
- The result is of higher type
- For example int ab float bc babc the resultant of abc expression is float Explicit Conversion It is the process in which we want to force a type conversion in a way that is different from the automatic conversion
- The general form of cast is (typename) expression Where typename is one of the standard C data type
- Implicit type declaration The general format is given below IMPLICIT type ( ) type ( )

file1_topic_52: precision, double, floatingpoint, point, decimal

- Float data type (for floatingpoint) floating point numbers are stored in bits with digit of precision
- Double data type (for double precision floating point number) the data type double is also used for handling floatingpoint numbers
- But it is treated as a distinct data type because it (double data type) occupies twice as much memory as type float and stores floatingpoint numbers with much larger range and precision
- There is one and only one decimal point
- This means that there must be one decimal point and there should not be more than one decimal points

file1_topic_53: character, compulsory, string, acquires, byte

- Char data type (for character) A single character can be defined as a character (char) type data
- Characters are usually stored in bits (one byte) of internal storage
- for character and string
- A string of character is called a character constant
- A variable which acquires only character string is called a character variable
- For character variables the declaration is compulsory

file1_topic_54: expression, evaluated, relation, conditional, exponentiation

- It is one if the specified relation is true and zero if the relation is false
- When postfix (or) is used with a variable in an expression the expression is evaluated first using the original value of the variable and then the variable is incremented (or decremented) by one
- expression expression The conditional expression is evaluated first
- If the result is nonzero expression is evaluated and is returned as the value of the conditional expression
- It return zero otherwise
- Any expression within parenthesis is first evaluated

- Exponentiation is given the top priority and evaluated first in an expression
- The value of the expression is evaluated first

file1_topic_55: rev, include, int, getch, hcf

- include include void main ( ) int abl printf (nEnter two numbers) scanf (ddab) labab printf (nThe largest number is d l) getch ( )
- (acac) (bcbc) printf (nThe largest value is dl) getch ( ) The goto statement It is the statement to branch unconditionally from one point to another in the program
- include include int hcf (int fint s) if (fs) return s return (sfs) void main ( ) int ab clrscr ( ) printf (nEnter the number) scanf (ddab) printf (The Hcf is d hcf (ab)) getch ( ) Storage classes The available storage classes are
- Reverse a number using recursive function include include void fib (int) void main ( ) int n rev (n) getch ( ) void rev (int n) static int r int d if (n) dn rrd nn rev (n) else printf (dr) Register variable We can tell the compiler that a variable should be kept in one of the machines registers instead of keeping in the memory (where normal variables are stored)
- include include void main ( ) int a int i clrscr ( ) for (iii) printf (nd u (ai) ai) getch ( ) Note

file1_topic_56: comma, expression, simple, combined, closes

- The comma operator The comma operator can be used to link the related expression together
- A comma linked list of expressions are evaluated left to right and the value of rightmost expression is the value of the combined expression
- The above statement can be equivalently written as int table Commas are require after each brace that closes off a row except in the case of the last row
- The expression may be any simple variable or structure variable or an expression using simple variables

file1_topic_57: output, blanks, carriage, unwanted, tabs

- This includes RETURN and TAB

- A dummy getchar ( ) may be used to eat the unwanted newline character
- We can also use the fflush function to flush out the unwanted character
- Blanks tabs or newlines
- Blanks tabs and newlines are ignored
- Carriage control In any output the first character of the output is lost
- That is considered as the carriage control
- So the first character of the output must be made a blank space so that the loss does not affect the output

file1_topic_58: string, arguments, control, argn, direct

- Control string and arguments are separated by commas
- Control string (also known as format string) contain field specification which direct the interpretation of input data
- argn) Control string consists of three types of items
- Escape sequence character such as nt and b The control string indicates how many arguments follow and what their types are
- The arguments argargargn are the variables whose values are formatted and printed according to the specifications of the control string

file1_topic_59: format, specification, specifications, sssr, denote

- It may include field (or format) specifications consisting of the conversion character a data type character (or type specifier) and an optional number specifying the field width
- Format specifications that define the output format for display of each item
- FORMAT specification When data are to be input or the result to be output we fully mention the type of the data (integer real or character) and also its size
- The specification of the types of the data and its size is called FORMAT specification
- FORMAT statement The general form of a FORMAT statement is n FORMAT (sssr) where n is the statement number sssr are the format specifications Rules
- I format The symbol I is used to denote the integer quantities
- The general I format specification is I w Where w is the width of the integer data
- Example This data can be describe by the format statement FORMAT (II) This can be also be written as FORMAT (I) F Format The symbol F is used to denote the real data

expressed in decimal form

file1_topic_60: break, skip, statement, continue, enclosing

- The break Statement The break statement enables a program to skip over part of the codes
- A break statement terminates the smallest enclosing while dowhile for or switch statement
- The continue Statement The continue is another jump statement some what like the break statement as both the statement skip over a part of the code
- But the continue statement is little different from break
- Instead of forcing termination it force the next iteration of the loop to take place skipping any code in between

file1_topic_61: referenced, hold, subscripts, indices, classroll

- cc Display Display Array Array is a collection of variables of the same type that are referenced by a common base
- The array is given a name and its elements are referred by their subscripts or indices
- The arrayname specifies the name with which the array will be referenced and size defined how many elements the array will hold
- array_nameimember_name Example classroll Practice Define a structure to hold the roll no

file1_topic_62: array, comprised, element, single, behaves

- Arrays are of different types (i) onedimensional array comprised of finite homogeneous element
- (ii) multidimensional array comprised of elements each of which is itself an array
- Single dimensional Array The simplest form of an array is a single dimensional array
- An array definition specifies a variable type and a name along with one more feature size to specify how many data items the array will contain
- The element value in the list of value must have the same data type as that of type of the array

- An array is a collection of related data element of same type
- Any array behaves like a builtin data type

file1_topic_63: initialization, size, initialized, dimensional, array

- Initialization of one dimensional array After an array is declared its elements must be initialized
- The general form of initialization of array is type arrayname size list of value The values in the list are separated by commas
- Following declares an int array sales of size int sale Initialization two dimensional array Two dimensional arrays are also initialize in the same ways as singledimension ones
- When the array is completely initialize with all values explicitly we need not specify the size of first dimension
- In such cases the size of array will be determined automatically based on the number of elements initialized

file1_topic_64: row, zero, remaining, elements, initialize

- Example int number If the number of initializers may be less than the declared size
- In such case the remaining elements are initialized to zero if the array type is numeric and NULL if the type is char
- Example int num Will initialize the first two elements of num as and respectively and the remaining elements to
- For example int table initializes the elements of the first row to zero and the second row to one
- The initialization is done row by row
- That is int table If the values are missing in an initialize they are automatically set to zero
- int table Will initialize the first two elements of the first row to one the first element of the second row to two and all other elements to zero
- It permitted to have a partial initialization
- We can initialize only the first few members and leave the remaining blank
- The uninitialized members should be only at the end of the list
- The uninitialized members will be assigned default values as follows Zero for integer and floating point numbers

file1_topic_65: string, char, include, printf, stringi

- include include void add (int nint r) int ij for (iii) ri for (jjj) rinij void main ( ) int listresultij printf (nEnter the number) for (iii) for (jjj) scanf (dlistij) add (listresult) printf (The result is) for (iii) printf (ndresulti) getch ( ) String C does not have a string data type rather it implements strings as single dimension character array
- Program to count number of word in a line
- include include include void main ( ) char string int ic clrscr ( ) printf (n Enter the first string) gets (string) istrlen (string) stringi for (istringii) if (stringi) c printf (n The number of word in sentencedc) getch ( ) Array of string An array of string is a twodimensional character array the size of first index (rows) determines the number of strings and the size of second index (column) determines maximum length of each string
- Pass these to a function which reverse the second string and then appends it at the end of the first string
- include include char rev (char name) int lj char r lstrlen (name) for (lljllj) rjnamel rj return r void display (char namechar name) char n nname This copy the name address to n printf (n the copy string is sn) namerev (name) strcat (nname) printf (n The new string issn) void main ( ) char stringstring clrscr ( ) printf (n Enter the first string) scanf (sstring) printf (n Enter the second string) scanf (sstring) display (stringstring) getch ( ) Structure A structure is a collection of variable (of different data types) referenced under one name providing of convenient means of keeping related information together

file1_topic_66: passed, array, passing, formal, argument

- Passing Array To Functions Three Rules to pass an array to a function
- Function must be called by passing only the name of the array
- In the function definition the formal parameter must be an array type the size of the array does not need to be specified
- The function prototype must show that the argument in a array
- Note When an entire array is passed as an argument the contents of the array are not copied into the formal parameter array
- Instead information about the addresses of array elements are passed on to the

function

- Therefore any changes introduced to the array elements are truly reflected in the original array in the calling function
- However this does not apply when an individual element is passed on argument
- The function must be called by passing only the array name

file1_topic_67: structure, argument, actual, copy, member

- We can not initialize individual members inside the structure template
- The order of values enclosed in braces must match the order of member in the structure definition
- The first method is to pass each member of the structure as an actual argument of the function call
- The general format of sending a copy of a structure to the called function as function_name (structure_variable_name) The called function takes the following form data_type function_name (Struct_type st_name) return (expression) The following points are important to note
- For example if it is returning a copy of the entire structure then it must be declared as struct with an appropriate tag name
- The structure variable used as the actual argument and the corresponding formal argument in the called function must be of the same struct type
- When a function returns a structure it must be assigned to a structure of identical type in the calling function

file1_topic_68: time, distance, struct, printf, sec

- Practice Create a structure containing real and imaginary as its member
- include include struct complex int r int m void main ( ) struct complex ccc printf (nEnter the real n imaginary for first complex) scanf (ddcrcm) printf (nEnter the real n imaginary for second complex) scanf (ddcrcm) crcrcrcmcm cmcrcmcmcr printf (nThe resultant is) printf (nReal is dcr) printf (nImaginary is dcm) getch ( ) Structure and function There are three methods by which the values of a structure can be transferred from one function to another
- Practice Create a structure TIME containing hour minutes and seconds as its member
- Write a program the uses this structure to input start time and stop time to a function

- Which returns the sum and difference of the start time and stop time in the main program
- include include struct time int hr int min int sec void input (struct time t) printf (nEnter the hr min and sec) scanf (dddthrtmintsec) void display (struct time t) printf (nhrdtmindtsecdtthrtmintsec) struct time add (struct time tstruct time t) struct time t tsectsectsec tmintmintmin thrthrthr tmintsec tsectsec thrtmin tmintmin return t struct time sub (struct time tstruct time t) struct time t tsectsectsec tmintmintmin thrthrthr if (tsec) tmin tsec if (tmin) thr tmintmin return t void main ( ) struct time ttt printf (nEnter the start time in hr min sec) input (t) printf (nEnter the stop time in hr min sec) input (t) tadd (tt) printf (naddition of two time is) display (t) tsub (tt) printf (nThe substraction of time is) display (t) getch ( ) Array of Structures We use structure to describe the format of a number of related variables
- Build functions to add and subtract given distances and display the results in the main function
- include include struct distance int feet int inch void input (struct distance d) printf (nEnter the feet and inches) scanf (dddfeetdinch) void display (struct distance d) printf (nFeetd t Inchedtdfeetdinch) struct distance add (struct distance dstruct distance d) struct distance d dfeetdfeetdfeet dinchdinchdinch dfeetdinch dinchdinch return d struct distance sub (struct distance dstruct distance d) struct distance d dinchdinchdinch dfeetdfeetdfeet if (dinch) dfeetdfeet dinchdinch return d void main ( ) struct distance ttt printf (nEnter the starting distance in feet and inches) input (t) printf (nEnter the stoping distance in feet and inches) input (t) tadd (tt) printf (naddition of two time is) display (t) tsub (tt) printf (nThe substraction of time is) display (t) getch ( ) Nested Structure Structure within a structure means nesting of structure

file1_topic_69: nonzero, file, pointer, specified, subsequently

- This pointer which contains all the information about the file is subsequently used as a communication link between the system and the program
- It take a FILE pointer as its only argument and returns a nonzero integer value if all the data from the specified file has been read and return a nonzero integer value if all of the data form the specified file has been read and return zero otherwise
- It fp is a pointer to file that has just been opened for reading then the statement
- This means that n bytes have already been read (or written)

file1_topic_70: rewind, file, remember, position, achieved

- In this mode FILE pointer points to the starting of bite of file
- Random access to files This can be achieved with the help of the function fseek ftell and rewind available in IO library
- Rewind takes a file pointer and resets the position to the start of the file
- rewind nd (fp) nfte (fp) Would assign to n because the file position has been set to the start of the file be rewind
- Remember the first byte in the file is numbered as second as and so on
- Remember that whenever a file is opened for reading or writing a rewind is done implicitly

file1_topic_71: error, fail, behave, typical, premature

- Error handling during IO operation It is possible that an error may occur during IO operations on a file
- Typical error situation include
- If we fail to check such read and write errors a program may behave abnormally when an error occurs
- An unchecked error may result in premature termination of the program or incorrect output

file1_topic_72: ferror, feof, message, error, end

- We have two statusinquiry library functions feof and ferror that can help up detect IO errors in the file
- The feof function can be used to test for an end of file condition
- if (feof (fp)) printf (End of data) would display the message End of data on reaching the end of file condition
- The ferror function report the status of the file indicated
- It also takes a FILE pointer as its argument and returns a nonzero integer if an error has been detected up to that point during processing
- The statement if (ferror (fp)) printf (An error has occurred) Would print the error message if the reading is no successful

file1_topic_73: goes, computed, nnnk, control, destination

- The general form is GO TO n where n is the statement number to which the control must be transferred
- The blank between GO and TO is optional
- GO TO In the GO TO statement the number referenced can not be a variable example GO TO I is not valid
- Computed GO TO statement The computed GO TO statement cause the transfer of control depending upon value of an integer variable
- the destination (where to go) is decided by the value in the integer variable
- the general form is GO TO (nnnk) i where i is the integer variable nnnk are statement numbers
- If the value is negative the control goes to statement number n if it is zero it goes to n and if it is positive it goes to n

file1_topic_74: implied, read, print, continue, loop

- CHARACTER statement Practice Write a program to find the range from the list of element using array in FORTRAN
- C RANGE OF MATRIX INTEGER A ( ) LS PRINT ENTER THE ELEMENT TO ARRAY DO I READ A (I) CONTINUE LA ( ) SA ( ) DO I IF (LLTA (I)) THEN LA (I) ENDIF IF (SGTA (I)) THEN SA (I) ENDIF CONTINUE PRINT LAGREST ELEMENTL PRINT SMALEST ELEMENTS PRINT RANGELS STOP END IMPLIED DO loop Suppose we want to read all the entries of a one dimensional array A with array length
- INTEGER A ( ) DO I READ A (I) CONTINUE FORTRAN also has the facility of reading or writing the entire array with one statement
- READ ( ) (A (I) I) Similarly the following is the implied DO loop to print the elements of the array
- C ASCENDING ORDER OF MATRIX INTEGER A ( ) TEMP PRINT ENTER THE ELEMENT TO ARRAY READ ( ) (A (I) I) DO I DO JI IF (A (I) GTA (J)) THEN TEMPA (I) A (I) A (J) A (J) TEMP ENDIF CONTINUE PRINT THE ELEMENT TO ARRAY WRITE ( ) (A (I) I) STOP END Implied DO loop for multidimensional array The implied DO loop can also be used for multidimensional arrays
- INTEGER A ( ) DO I DO J READ A (IJ) CONTINUE But we can use the implied DO loop

and write this as a single statements as follows

- READ ( ) ((A (IJ) J) I) Notice that the outer loop has I as running variable and the inner loop has J

file2_topic_75: problem, solving, steps, solve, analyse

- Problem is defined as the difference between an existing situation and a desired situation that is in accordance with calculation a problem is numerical situation and has complex form
- If a problem is solved by computing using machine called computer then such process is called Problem Solving using Computer
- Problem Analysis If you have studied a problem statement then you must analyse the problem and determine how to solve it
- First you should know the type of problem that is nature of problem
- At first you try to solve manually
- If it is solvable manually by using your idea and knowledge then you can use such idea and principle in programming and solve the problem by using computer
- So you must have well knowledge about a problem
- In order to get exact solution you must analyse the problem
- To analyse means you should try to know the steps that lead you to have an exact solution
- Suppose you are asked by your father to solve an arithmetic problem and you are not familiar with the steps involved in solving that problem
- In such a situation you will not be able to solve the problem
- Suppose you know the steps to be followed for solving the given problem but while solving the problem you forget to apply some steps or you apply the calculation steps in the wrong sequences
- What is a Problem
- Why do we need documentation and testing for problem solving
- Generally a difficult problem is divided into sub problems and then solved
- The solution of the problem is built on the results from the smaller versions
- They are Problem could be written or defined in terms of its previous result

file2_topic_76: errors, wrong, computer, programmers, occur

- In programming point of view the problem must be computing
- Obviously you will get a wrong answer
- Similarly while writing a computer program if the programmer leaves out some of the instructions for the computer or writes the instructions in the wrong sequences then the computer will calculate a wrong answer
- Generally programmers commit three types of errorsThey are
- Runtime errors Syntax errors are those errors which are arised from violating the rules of programming languageOn encountering these errors a computer displays error messageIt is easy to debugLogic errors are those which arised when programmers proceed the logic process in wrong way or miss the some statementsIt is difficult to debug such errors because the computer does not display themRuntime errors are those which occur when programmers attempt to run ambiguous instructionsThey occur due to infinte loop statementdevice errorssofware errors etcThe computer will print the error message Some of runtime errors are Divide by zero Null pointer assignment Data over flow Testing is the process of reviewing and executing a program with the intent of detecting errors
- The purpose of this material is to introduce certain basic concept and to provide some necessary definitions

file2_topic_77: instruction, computer, programming, sequence, language

- The same principle applies to writing computer program also
- A programmer can not write the instruction to be followed by a computer unless the programmer knows how to solve the problem manually
- Thus to produce an effective computer program it is necessary that the programmers write each and every instruction in the proper sequence
- However the instruction sequence (logic) of a computer program can be very complex
- Hence in order to ensure that the program instructions are appropriate for the problem and are in correct sequence program must be planned before they are written
- Coding In order to make a program in any programming language what we have written is known as code
- The act of writing code in a computer language is known as coding
- In other words code is a set of instruction that a computer can understand
- Compilation Execution The process by which source codes of a computer (

programming) language are translated into machine codes is known as compilation
- A programming language is designed to help process certain kinds of data consisting of numbers characters and strings and to provide useful output known as information
- The task of programming of data is accomplished by executing a sequence of precise instruction called a program
- Normally however this is not done as it is considered a poor programming practice

file2_topic_78: algorithm, characteristics, posses, prepare, formally

- Algorithm Development Flowcharting The term algorithm may be formally defined as a sequence of instructions designed in such a way that if the instructions are executed in the specified sequence the desired result will be obtained
- An algorithm must posses the following characteristics
- Normally an algorithm is first represented in the form of a flowchart and the flowchart is then expressed in some programming language to prepare a computer program
- Define algorithm and flowchart
- Differentiate between the flow chart and algorithm with the example

file2_topic_79: instruction, infinitely, noted, unambiguous, strictly

- Each and every instruction should be precise and unambiguous
- Each instruction should be such that it can be performed in a finite time
- One or more instruction should not be repeated infinitely
- It may also be noted that in order to solve a given problem each and every instruction must be strictly carried out in a particular sequence

file2_topic_80: students, examination, roll, obtained, subjects

- Problem There are students in a class who appeared in their final examination
- Their mark sheets have been given to you
- Problem A student appears in an examination that consists of total subjects each subject having maximum marks of
- The roll number of the students his name and the marks obtained by him in various subjects is supplied as input data

- Draw a flowchart for the algorithm to calculate the percentage marks obtained by the student in this examination and then to print it along with his roll number and name
- Display the name and roll no of those students mark is greater than
- Write a program to read several different names roll address percentage and display name who has score the rd highest
- Feed the marks obtained by three students in each subjects and calculate the total of each student
- Write a program to enter name roll and mark of students and store them in the file
- (PU) Write a program to create a data files containing record roll number and students name and total marks

file2_topic_81: step, total, mark, division, sheet

- Write an algorithm to calculate and print the total number of students who passed in first division
- Algorithm Step Initialize Total First Division and Total Mark sheet checked to zero ie
- total_first_div total_marksheet_chkd Step Take the mark sheet of the next student
- Step Check the division column of the mark sheet to see if it is I if no go to step
- Step Add to Total First Division ie
- total_first_div Step Add to Total Mark sheets checked ie
- total_marksheet_chkd Step Is Total Mark sheets checked if no go to step Step Print Total First Division

file2_topic_82: flowchart, logic, pictorial, drawing, flow

- Flowchart A flowchart is a pictorial representation of an algorithm that uses boxes of different shapes to denote different types of instructions
- The main advantage of this two steps approach in program writing is that while drawing a flowchart one is not concerned with the details of the elements of programming language
- Since a flowchart shows the flow of operations in pictorial form any error in the logic of the procedure can be detected more easily than in the case of a program
- Once the flowchart is ready the programmer can forget about the logic and can concentrate only on coding the operations in each box of the flowchart in terms of the statements of the programming language

- A flowchart therefore is a picture of the logic to be included in the computer program
- Experienced programmers sometimes write programs without drawing the flowchart
- However for a beginner it is recommended that a flowchart be drawn first in order to reduce the number of errors and omissions in the program
- It is a good practice to have a flowchart along with a computer program because a flowchart is very helpful during the testing of the program as well as while incorporating further modifications in the program
- The communication of program logic through flowcharts is made easier through the use of symbols that have standardized meanings
- The normal flow of flowchart is from top to bottom and left to right
- Connector If a flowchart becomes very long the flowlines start crisscross at many places that causes confusion and reduces understandability of the flowchart
- What is flow charting describe its importance
- What is a flowchart

file2_topic_83: flowlines, symbols, arrowheads, flowchart, good

- Flowchart Symbols A flowchart uses boxes of different shapes to denote different types of instructions
- Only a few symbols are needed to indicate the necessary operations in a flowchart
- Flowlines Flowlines with arrowheads are used to indicate the flow of operations that is the exact sequence in which the instructions are to be executed
- Arrowheads are required only when the normal top to bottom flow is not to be followed
- However as a good practice and in order to avoid ambiguity flowlines are usually drawn with an arrowhead at the point of entry to a symbol
- Good practice also dictates that flowlines should not cross each other and that such intersections should be avoided whenever possible
- What are the various symbols used to a flowchart
- List the various commonly used flowchart symbols

file2_topic_84: true, operands, logic, result, false

- For example a diamond always means a decision
- it causes an expression that is originally true to become false and viceversa

- The result of a logic and operation will be true only if both operands are true where as the result of a logic or operation will be true if either operand is true or if both operands are true
- In other word the result of a logic or operation will be false only if both operands are false

file2_topic_85: inputoutput, input, terminal, device, output

- These symbols are listed below Terminal InputOutput Processing Terminal The terminal symbol as the name implies is used to indicate the beginning (START) ending (STOP) and pauses (HALT) in the program logic flow
- InputOutput The inputoutput symbol is used to denote any function of an inputoutput device in the program
- If there is a program instruction to input data from a disk tape cardreader terminal or any other type of input device that step will be indicated in the flowchart with an inputoutput symbol
- Similarly all output instructions whether it is output on a printer magnetic tape magnetic disk terminal screen or any output device are indicated in the flowchart with an inputoutput symbol
- The data that is given to a program is known as input data
- Similarly the data that is provided by a program is known as output data
- Generally input data is given to a program from a keyboard (a standard input device) or a file
- The program then proceeds the input data and the result is displayed on the screen (monitor a standard output device) or a file
- Reading input data from keyboard and displaying the output data on screen such input output system is considered as conio input out
- Input Output Fig Input Output To perform inputoutput operation in console mode C has a number of input and output functions
- When a program needs data it takes the data through the input functions and sends the results to output devices through the output functions
- Thus the inputoutput functions are the link between the user and the terminal
- As keyboard is a standard input device the input functions used to read data from keyboard are called standard input functions
- So its work is merely to hold the output screen until a key is pressed

- Describe any two file handling inputoutput function

file2_topic_86: executed, order, consecutively, enable, computations

- When more than one arithmetic and data movement instructions are to be executed consecutively they are normally placed in the same processing box and they are assumed to be executed in the order of their appearance
- In the absence of control statements the instruction or statements are executed in the same order in which they appear in the program
- So control statements enable use to specify the order in which various instruction in the program are to be executed
- This process is used for repetitive computations in which each action is stated in term of previous result
-) Increasing the execution speed as they refer address

file2_topic_87: testing, based, inspection, manual, codes

- Testing can be done manually and computer based testing
- Manual Testing is an effecting errordetection process and is done before the computer based testing begins
- Manual testing includes code inspection by the programmer code inspection by a test group and a review by a peer group
- Computer based testing is done by computer with the help of compiler (a program that changes source codes into machine codes word by word)

file2_topic_88: documentation, internal, define, explanatory, comments

- Program Documentation Program Documentation refers to the details that describe a programWhile writng programs it is good programming practice to make a brief explanatory note on the programor program segmentThis explanatory note is called commentIt explains how the program works and interact with itThus it helps other programmers to understand the programThere are two types of documentation Internal documentation Some details may be builtin as an integral part of the program
- These are known as internal documentation

- Two important aspects of internal documentation are selection of meaningful variable names and the use of comments
- For example Area Breadth Length is more meaningful than A B L And comments are used to describe actions parts and identification in a program
- External documentation is an executable statement in a programIt may be message to the user to respond to the program requirementThis is accomplished using output statements It makes a program more attractive and interactiveSome examples are print Input the number one by one print Do you want to continue Some Important Questions
- This section consists two parts declaration part and executable part
- For example define TAXRATE define PI define TRUE define FALSE define FRIEND Susan area PI radius radius is equivalent to Some Important Questions
- It can also be defined as a section of a program performing a specific task
- It may be auto static extern and register It is optional

file2_topic_89: unix, operating, developed, language, level

- C contains additional features that allow it to be used at a lower level thus bridging the gap between machine language and the more conventional high level language
- This flexibility allows C to be used for system programming (eg
- The new language was named C Since it was developed along with the UNIX operating system it is strongly associated with UNIX
- This operating system was developed at Bell Laboratories and was coded almost entirely in C C was used mainly in academic environments for many years but eventually with the release of C compiler for commercial use and the increasing popularity of UNIX it began to gain widespread support among compiler professionals
- Today C is running under a number of operating systems including MsDOS
- In fact many of the C compilers available in the market are written in C Programs written in C are efficient and fast
- C is highly portable
- This means that C programs written for one computer can be seen on another with little or no modification
- Portability is important if we plan to use a new computer with a different operating system
- Another important feature of C is its ability to extend itself

- These computers often have small amounts of storage within the CPU itself where data can be stored and accessed quickly

file2_topic_90: language, suited, programming, high, builtin

- It also resembles other high level structure programming language such as Pascal and Fortran
- B language was modified by Dennis Ritchie and was implemented at Bell Laboratories in
- It is a robust language whose rich set of builtin functions and operators can be used of builtin functions and operators can be used to write any complex program
- The C compiler combines the capabilities of an assemble language with the features of a highlevel language and therefore it well suited for writing both system software and business packages
- It is many times faster than BASIC (Beginners All Purpose Symbolic Instruction Code a high level programming language)
- C Language is well suited for structure programming thus requiring the user to think of a problem in terms of function modules or blocks

file2_topic_91: collection, functions, modules, divided, task

- A proper collection of these modules would make a complete program
- A C program is basically a collection of functions that are supported by the C library
- With the availability of a large number of functions the programming task becomes simple
- Basic Structure of C programs Every C program consists one or more modules called function
- Every C program can be though of as a collection of these functions
- Each program has one or more functions
- A program can be divided into functions each of which performs some specific task
- So the use of C functions modularizes and divides the work of a program
- Types of Functions C program has two types of functions
- A complex C program can be divided into a number of user defined functions

file2_topic_92: global, variables, local, outside, life

- Such variables are called global variables and are declared in global declaration section that is outside of all the function
- The local variables are created when the function is called and destroyed automatically when the function is exited
- Life time The period of time during which memory is associated with a variable is called the Program to illustrate local variable long int fact (int n) int i long int f for (i in i) f i return (f) main ( ) int num clrscr ( ) printf (Enter a number) scamf (d num) printf (The factorial of d is ld num fact (num)) Output Enter a number The factorial of is Global Variables (External) The variables that are defined outside any function are called global variables
- All functions in the program can access and modify global variables
- IT is useful to declare a variable global if it is to be used by many functions in the program
- The life time is as long as the programs execution does not come to an end
- What are local and global variables

file2_topic_93: semicolon, executable, declaration, variables, declares

- The declaration part declares all the variables used in the executable part
- All the statements in the declaration and executable parts ends with a semicolon
- The type declaration will apply throughout the program
- Declarations A declaration associates a group of variables with a specific data type
- All variables must be declared before they can appear in executable statements
- A declaration consists of a data type followed by one or more variable names ending with a semicolon
- The userdefined data type identifier can later be used to declare variables
- It doesnt require semicolon

file2_topic_94: braces, unit, closing, opening, surrounded

- These two parts must appear between the opening and the closing braces
- The program execution begins at the opening braces and ends at the closing brace
- This is a logical unit composed of a number of statements grouped into a single unit
- It is a block of statements surrounded by braces

file2_topic_95: exit, jump, brace, demonstrate, inclusion

- The closing brace of the main ( ) function section is the logical end of the program
- Exit ( ) function We have already known that we can jump out of a loop using either the break statement or goto statement
- In a similar way we can jump out of a program by using the library function exit ( )
- The general syntax is if (condition) exit ( ) The exit ( ) function takes an integer value as its argument
- The use of exit ( ) function requires the inclusion of the header file
- Program to demonstrate exit ( ) include include main ( ) int choice clrscr ( ) while ( ) printf (

file2_topic_96: opening, establishing, labc, file, connection

- The file name should end with the characters c like program c labc etc
- This provides the operating system the name of the file and the mode in which the file is to be opened
- The process of establishing a connection between the program and file is called opening the file
- In other word it specifies the purpose of opening a file
- Explain in brief the steps involved in opening a file

file2_topic_97: constant, set, character, equivalent, characters

- Character Set C uses the uppercase letters A to Z the lowercase letters a to z the digits to and certain special characters as building blocks to form basic program elements (eg
- Several character constants are shown below A X Character constants have integer values that are determined by the computers particular character set
- Thus the value of a character constant may vary from one computer to another
- The constants themselves however are independent of the character set
- This feature eliminates the dependence of a C program on a particular character set
- American Standard Code for Information Interchange) character set in which each individual character is numerically encoded with its own unique bit combination (hence

a total of difference characters)
- Several character constant and their corresponding values as defined by ASCII Constant Value A X These values will be the same for all computer that utilize the ASCII character set
- String Constants A string consists of any number of consecutive characters (including none) enclosed in (double) quotation marks
- A character constant (eg
- A) and the corresponding singlecharacter string constant (A) are not equivalent
- A character constant has an equivalent integer value whereas a single character string constant does not have an equivalent integer value and in fact consists of two characters the specified character followed by the null character (o)
- The null character constant is not equivalent to the character constant
- The characters may represent numeric constant a character constant or a string constant
- What do you mean by Character set in C
- A symbolic constant NULL is defined in stdioh which denotes the value zero

file2_topic_98: whitespace, groups, group, characters, multiple

- (Blank space) (Horizontal tab) (White Space) Most versions of the language also allow certain other characters such as and to be included with strings comments
- Must not contain white space
- The control string consists of individual groups of characters with one character group for each input data item
- Within the control string multiple character groups can be contiguous or they can be separated by whitespace (ie
- blankspace) tabs or newline characters
- If whitespace characters are used to separate multiple character groups in the control string then all consecutive whitespace characters in the input data will be read but ignored
- The use of blank spaces as character group separators is very common
- The consecutive nonwhitespace characters that define a field
- Multiple character group can be contiguous or they can be separated by other characters including whitespace character
- The use of blank spaces are character group separators is particularly common

file2_topic_99: tokens, linefeed, token, punctuation, passage

- Identifiers Keywords C Tokens In a passage of text individual words and punctuation marks are called tokens
- Similarly in a C program the smallest individual units are also known as C tokens
- C has six types of tokens
- For example a linefeed (LF) which is referred to as a newline in C can be represented as n
- What are the token of C language

file2_topic_100: loop, continue, rem, body, sum

- int float for while
- Stop Counter Initialization Test Condition False True Body of Loop Update expression Out of Loop For example Calculate the factorial of a number factorialc include main ( ) int num i long fact clrscr ( ) printf (n Enter a number whose factorial is to be calculated) scanf (d num) for (i inum i) fact i fact facti printf (n The factorial is d fact) getch ( ) Output Enter a number whose factorial is to be calculated The factorial is While Loop The while statement can be written as while (condition) while (condition) statement statement body of the loop statement First the condition is evaluated if it is true then the statements in the body of loop are executed
- fig Flowchart of while loop Program to print the sum of digits of any num while condition Body of Loop Next statement out of loop include main ( ) int n sum rem clrscr ( ) printf (Enter the number) scanf (d n) while (n) rem n taking last digit of number sum rem sum sum rem n n n skipping last digit end of while printf (Sum of digits d n sum) getch ( ) Output Enter the number Sum of digits dowhile loop The dowhile statement is also used for looping
- program to print the number from to using do while include main ( ) int i clrscr ( ) do printf (dt i) i while (i) printf (n) getch ( ) Output Body of Loop False Condition Next statement out of loop True Differences between while loop and do while loop Nesting of loops When a loop is written inside the body of another loop then it is known as nesting of loops
- The general syntax continue fig flowchart of continue control statemen Program to demonstrate continue statement include main ( ) int i num clrscr ( ) printf (n Enter a

number) scanf (d num) printf (n The even numbers from to d are n num) for (i inum i) if (i) continue printf (td i) end of for loop getch ( ) end of main ( ) Output Enter a number The even numbers from to are Loop statements before continue Next iteration of loop condition continue statements True false Loop statements after continue goto statement The goto statement is used to alter the normal sequence of program execution by unconditionally transferring control to some other part of the program
- t) scanf (d n) printf (n Enter d numbers n n) for (i inumj) temp numi numi numj numj temp end of if end of inner loop end of outer loop printf (n The numbers in ascending order n) for (i in i) printf (td numi) getch ( ) end of main Output How many numbers are there

file2_topic_101: lowercase, letters, upper, uppercase, letter

- Identifiers consisted of letters and digits in any order except that first character must be a letter
- Both upper and lower case letters are permitted though common usage favors the use of lowercase letters for most type of identifiers
- Upper and lowercase letters are not interchangeable (ie
- an uppercase letter is not equivalent to the corresponding lowercase letters)
- Must consist of only letters digits or underscore
- Only first characters are significant
- Since upper and lowercase characters are not equivalent it is possible to utilize an uppercase keyword as an identifier
- It can then be followed by any combination of digits taken from the sets through and a through f (either upper or lower case)
- The letters may be written in either upper or lowercase
- Since the first lowercase letter (in this case a) would be interpreted as the first character beyond the string
- Write a program to convert a lowercase character string into uppercase using array
- For example fclose (fp) fclose (fp)

file2_topic_102: underscore, middle, asterisk, period, dot

- The underscore (_) can also be included and considered to be a letter
- An underscore is often used in middle of an identifier

- An identifier may also begin with an underscore
- To do so the sign within the appropriate control group is followed by an asterisk ( )
- It must start with a period ( )
- The dot ( )

file2_topic_103: illegal, exponent, character, blank, xbff

- First character must be an alphabet (or Underscore)
- Illegal character (blank space) Illegal character ( ) the first digit can not be zero
- Illegal character ( ) Illegal character ( )
- xbff Illegal character ( )
- Either a decimal point or an exponent must be present
- Illegal character ( ) E The exponent must be an integer (it can not contain a decimal point) E Illegal character (blank space) in the exponent
- However will not be assigned partno because of asterisk which is interpreted as an assignment suppression character

file2_topic_104: bytes, byte, signed, unsigned, needs

- Data Types C language is rich in its data types
- C supports several different types of data each of which may be represented differently within the computer memory
- Typical memory requirements are also given Data Types Description Typical Memory Requirement char single character byte int integer quantity bytes float floatingpoint number bytes (word) double doubleprecision floating point number bytes (word) In order to provide some control over the range of numbers and storage space C has following classes signed unsigned short long
- Types Size char or signed char byte unsigned char byte int bytes short int byte unsigned short int byte signed int bytes unsigned int bytes long int bytes signed long int bytes unsigned long int bytes float bytes double bytes long double bytes void is also a builtin data type used to specify the type of function
- Because I needs bytes f needs bytes and c needs byte bytes is highest in these

file2_topic_105: union, structure, created, keyword, space

- structure union enum The basic data types are also known as built in data types
- is i Multiplication is i Union and its important Union is similar to structure data type but union store value of different types in a single location
- Union will contain many different type of values but only one is stored at a time
- The declaration of union is same as the structure
- Only difference is in place of struct keyword the union keyword is used
- The accessing of union member is also same like structure member
- For example Union data int a float b The union variable are declared like structure variable
- For example Union data d d If we write the statement da Then that means the field a is valid but we want to print the value of b after above assignment the value is wrong because a is valid only
- Space Created for Union We know that union store one value at a time so the question is now much space is created for union
- For example Union A int i float f char c union A a The space is created for a is byte
- Distinguish between Structure and union

file2_topic_106: octal, digits, taken, combination, digit

- Thus it consists of a sequence of digits
- A decimal integer constant can consists of any combination of digits taken from the set through
- An octal integer constant can consist of any combination of digits taken from the set through
- However the first digit must be in order to identify the constant as an octal number
- The general form represents an octal digit (through)

file2_topic_107: constants, illegal, integer, incorrectly, reason

- Several valid decimal integer constants are shown below The following decimal integer constants are written incorrectly for reason stated Illegal character ( ) Illegal character ( )
- Several valid hexadecimal integer constants are shown below x X XFFF xabcd The following hexadecimal integer constants are written incorrectly for the reason stated X Illegal character ( )

- XDEFG Illegal character (G) Unsigned and Long Integer Constants Unsigned integer constants may exceed the magnitude of ordinary integer constants by approximately a factor of l though they may not be negative
- The constants following the case keywords should be of integer or character type

file2_topic_108: null, termination, terminating, string, length

- Thus the string would be displayed as Line Line Line The compiler automatically places a null character ( ) at the end of every string constant as the last character within the string (before the closing double quotation mark)
- Normally zero is used to indicate normal termination and non zero value to indicate termination due to some error or abnormal condition
- In other words we can say that a string is a sequence of contiguous character in memory terminated by the null character
- The terminating null character is important because it is the only way the string handling functions can know there the string ends
- The second dimension tells the maximum length of each string
- In above declaration we can store strings each can store maximum characters last th space is for null terminator in each string
- The length of string is the number of characters present in it excluding the terminating null character

file2_topic_109: yielding, integervalued, represent, integer, returning

- Thus a b and c will each represent an integervalued quantity and d will represent a single character
- all integers all characters)
- For example integer number ie
- It can be value of any integer or character variable or a function call returning on integer or an arithmetic logical relational bitwise expression yielding integer value

file2_topic_110: assigned, follow, quantity, value, values

- The next four lines cause the following things to happen the integer quantity is

assigned to a is assigned to b and the quantity represented by the sum ab (e
-) is assigned to c The character a is assigned then assigned to d In the third line within this group the values of the variables a and b are accessed simply by writing the variables on the righthand side of the equal sign
- The last four lines redefine the values assigned to the variables as the integer quantity is assigned to a replacing the earlier value then is assigned to b replacing the earlier value The difference between a and b (ie
- A comma linked list of expression are evaluated left to right and the value of rightmost expression is value (x y x y) Here is assigned to x and is assigned to y and so expression xy is evaluated as ( ) ie
- a b In this example x will be assigned the value of b
- int subject char sex M F float marks int element In example ( ) elements are five but we are assigning only three values
- int) The value is assigned like follow a a a a a a int a In above example the two value in the first inner pair of braces are assigned to the array element in the first row the values in the second pair of braces are assigned to the array element in the second row and so on
- The value assign like follow a a a a a a In above example the value zero is assigned to a and a because no value assigned to these
- In above example value assign like follow a a a a a a int a int a would never work

file2_topic_111: size, array, dimension, tells, elements

- Each array variable must be followed by a pair of square brackets containing a positive integer which specifies the size (ie
- the number of elements) of the array
- The size of this array will be equal to the value of argc
- size of the array is the number of elements in the array
- For example define size a size The size of the array must be specified ie
- For example char str The first dimension (size) tells how many strings are in the array

file2_topic_112: california, text, meaningless, extra, filled

- char text California This declaration will cause text to be an element character array

- The first elements will represent the characters within the word California and the th element will represent the null character ( ) which automatically added at the end of the string
- char text California the character at the end of the string (in this case the null character) will be lost
- char text California the extra array elements may be assigned zeros or they may be filled with meaningless characters

file2_topic_113: items, consider, type, data, integers

- The data items must all be of the same type (eg
- For example consider the following data
- In such situation where we have multiple data items of same type to be stored we can use array
- The individual data items can be characters integers floating point numbers etc
- However they must all be of the same type and the same storage class (ie
- The data item can be different type some can be int some can be float some can be char and so on

file2_topic_114: escape, sequence, newline, feed, ascii

- An escape sequence always begins with a backslash and is followed by one or more special characters
- Such escape sequences always represent single characters even though they are written in terms of two or more characters
- The commonly used escape sequences are listed below Character Escape Sequence ASCII Value bell (alert) a backspace b horizontal tab t vertical tab v newline (line feed) n form feed f carriage return r quotation mark ( ) apostrophe ( ) question mark ( )
- Escape Sequence represents the null character (ASCII) which is used to indicate the end of a string
- The general form of a hexadecimal escape sequence is xhh where each h represents a hexadecimal digit (through and a through f)
- What is Escape Sequence
- Write any four escape sequence with meaning and symbols
- List out any threeescape sequence with their uses

- If the character within the brackets is simply the circumflex followed by a newline character then string entered from the standard input device can contain any ASCII characters except the newline characters (line feed)
- The Enter Key will issue the newline character thus signifying the end of the string

file2_topic_115: typedef, type, units, existing, identifier

- Typedef Statement C supports a feature known as type definition that allows users to define an identifier that would represent an existing data type
- It takes the general form typedef type identifier where type refers to an existing data type and identifier refers to the new name given to the data type
- typedef can not create a new type
- Some examples of type definition are typedef int units typedef float marks where units represent int and marks represents float
- The main advantage of typedef is that we can create meaningful data type names for increasing the readability of the program

file2_topic_116: cprogram, kumar, ram, cprogramming

- Describe different data types are used in cprogram
- Write down the various types of it in Cprogram
- a include include void main ( ) char name clrscr ( ) printf (Enter your name) gets (name) printf (your name is) puts (name) getch ( ) output Enter your name Ram Kumar Your name is Ram Kumar Some Important Questions Write a general form of inputoutput statement of Cprogramming
- CPROGRAM X_FILE Y_FILE) is passed on to the program through these arguments when main ( ) is called up by the system
- How and when the they declared in Cprogram

file2_topic_117: division, integer, respectively, note, perform

- Division of one integer quantity by another is referred to as integer division
- Suppose that i j and k are integer variables whose values are and respectively
- If the first number is greater than second perform multiplication otherwise division

- Note that x and y both are integer type
- Where x and y both are integer type

file2_topic_118: floatingpoint, precision, variable, hand, dropped

- the decimal portion of the quotient will be dropped)
- On the other hand if a division operation is carried out with two floatingpoint numbers or one floating point numbers and other integer the result will be a floating point
- Suppose that i is an integer variable whose value is f is a floatingpoint variable whose value is and c is a character variable that represents the character w
- Suppose that i is an integer variable x is a floatingpoint variable d is doubleprecision variable and c is character type variable
- The first be read into a variable int the second into float and the third into char
- The second will be assigned to a long decimal integer variable and the third will be assigned to a double precision variable
- ¾ Precision Optional The operation of precision field depends on the types of conversion

file2_topic_119: conditional, expressions, interpretation, make, logical

- Several relational expressions involving these variables Expressions Interpretation Value i k true (jk) (i) false k
- not Logical operators are used to compare evaluate logical and relational expressions
- Several complex logical expressions that make use of these variables are shown below Expression Interpretation Value (i) (c w) true (i) (c) true (f) false (c
- is known as conditional operator
- Simple conditional operations can be carried out with conditional operator
- An expression that make use of the conditional operator is called a conditional expression
- Make the comparison between them with conditional operator

file2_topic_120: expression, conditional, evaluating, negates, value

- that negates the value of a logical expression ie

- p) (if) true
- expression expression When evaluating a conditional expression expression is evaluated first
- If expression is true the value of expression is the value of conditional expression
- If expression is false the value of expression is the value of conditional expression

file2_topic_121: operand, postfix, prefix, altered, utilized

- If the operator is written before the operand then it is called as prefix unary operator
- If the operator is written after the operand then it is called as postfix unary operator
- When prefix is used the operand will be altered in value before it is utilized for its intended purpose within the program
- Similarly when postfix is used the operand will be altered in value after it is utilized
- It is a compile time operand
- The operand may be a variable a constant or a data type qualifier
- This operator can be read as the address of so sn means address of sn similarly price means address of price

file2_topic_122: printf, gender, output, str, format

- Suppose the program includes the following three printf statements
- The statements printf (integer d n size of i) printf (float d n size of x) printf (double d n size of d) printf (character d n size of c) might generate the following output integer float double character The above statement can also be written as printf (integer dnsizeof (int)) and so on
- char text Kathmandu printf (Number of characters dsizeof (text)) will generate the following output Number of characters Conditional Operators The operator
- The printf ( ) is a builtin function which is used to output data from the computer onto a standard output device ie
- The printf ( ) statement provides certain features that can be used to control the alignment and spacing of printouts on the terminals
- The general form of printf ( ) statement is printf (control string arg arg
- So the output generated will be fastener Example include main ( ) double x y printf (fffffnn x y xy xy) printf (eeee x y xy xy) The output are e e e Example read and write a line of text include main ( ) char line scanf (n line) printf (s line) Arun Kumar Arun

Kumar Control string The general syntax of control string Flag Field width precision conversion character ¾ Flags Optional T he flag affect the appearance of the output
- int n Format Output l w printf (d n) wl printf (d n) printf (d n) wl printf (d n) Format for floating point output The general form wpf wpe where w is the integer width including decimal point p is the precision f and e are conversion characters Example Format Output printf (d) printf ( ) printf (x) c printf (f)
- e printf (e) printf (g) eg
- float x Format Output printf (f x)
- printf (f x)
- printf (f x) _ printf (f x) printf (f x) printf (e x) e printf (e x) e printf (e x) e _ Output of Strings The general form of control string is wps where w specifies the field width for display and p instructs that only the first P characters of the string are to be displayed
- Format Output M Y N E P A L printf (s str) M Y N E P A L printf (s str) M Y N printf (s str) M Y N E P printf (s str) printf (s str) M Y N E P A L Program to demonstrate unformatted function unformatc include include main ( ) clrscr ( ) char gender printf (Enter gender M or F) putchar (gender) getch ( ) Output Enter gender M or F M Our gender is M Note clrscr ( ) is a console fumction used to clear console (display) screen clrscr ( ) is pronoused clearscreengetch ( ) function is used to hold the console screen
- (PU) Find the output the following program (PU) include Main ( ) int abc a b c printf (dddnabc) printf (dddnabc) Find the output of the following program (PU) include include void main ( ) int x float y char z C clrscr ( ) printf (d f c nxyz) getch ( ) Write short noteson Formatted inputoutput function
- (PUBack) b Find the output of the following program
- include include main ( ) int a int P i clrscr ( ) p a printf (dn (pi)) getch ( ) The output of above program is Because the statement printf (dn (pi)) exzecute times from i to i
- Similarly other elements are printed
- EOF) printf (sf in std

file2_topic_123: bitwise, shift, bits, operators, operator

- Bitwise Operators Bitwise Operators are used for manipulating data at bit level
- These operators are used for testing the bits or shifting them right or left
- Bitwise operators can be applied only to Operator Meaning Bitwise AND Bitwise OR Bitwise exclusive OR Right shift Bitwise ones complement operator Consider a and b

- THe binary representation of a and b for bits a b c a b d a b n a e a b For Bitwise Shift Operator Operand Bitwise Shift operator number For eg

file2_topic_124: bytes, byte, address, allocated, memory

- Size of Operator The size of operator is used with an operand to return the number of bytes it occupies
- Ram The above line contains three types of data and must be read according to its format
- It is assumed that the starting memory location is
- As each integer element requires bytes subsequent element appears after gap of locations
- Normally each character is stored in one byte and successive characters of the string are stored in successive bytes
- The memory in a computer is made up of bytes arranged in a sequential manner
- Each byte has an index number which is called address of that byte
- The address of these bytes start from zero and the address of last byte is one less than the size of memory
- Suppose we have MB of RAM (Random Access Memory) then memory will consist of the address of these bytes will be from to
- The address of first byte from the two allocated bytes is ka the address of variable age
- Suppose compiler has reserved bytes numbered and for the storage of variable age then the address of variable age will e
- Memory Address age Now this value will be stored in these bytes in form of binary representation
- The number of bytes allocated will depend on the data type of variable
- For example bytes would have been allocated for a float variable and the address of first byte would be called the address of variable
- Generally bytes are used to store an address (may vary in different computers) so the compiler allocates bytes for a pointer variable
- bytes bytes) is reserved and the address of the first byte of the memory allocated is assigned to the pointer x of type int (ie
- x refers the first address of allocated memory)

file2_topic_125: check, img, odd, number, num

- Write a program to enter two numbers
- For eg Program to check whether the number is ve or ve include main ( ) int num clrscr ( ) printf (Enter a number to be tested) scanf (d num) if (num) printf (The number is negative) printf (value of num is dn num) getch ( ) Output st run Enter a number to be tested The number is negative Value of num is nd run Enter a number to be tested Value of num is True Condition Statement False Next Statement ifelse statement The ifelse statement is an extension of the simple if statement
- Program to check whether the number is even or odd include main ( ) int num remainder clrscr ( ) printf (Enter a number) Condition statement statement Next statement scanf (d num) remainder num modular division if (remainder) test for even printf (Number is evenn) else printf (Number is oddn) getch ( ) Output Enter a number Number is odd
- Program to print whether the number is even or odd include main ( ) int n clrscr ( ) printf (Enter the number) scanf (d n) if (n) goto even else goto odd even printf (Number is even) goto end odd printf (Number is odd) end printf (n) getch ( ) Output Enter the number Number is even
- Display all recordsn) printf (Enter your choice) scanf (d choice) switch (choice) case printf (datase created nn) break case printf (Record inserted nn) break case printf (Record modified nn) break case printf (Record deleted nn) break case printf (Record displayed nn) break case exit ( ) default printf (Wrong choicen) end of switch end of while getch ( ) end of main ( ) s Some Others Programs QWrite a program to check whether the number is prime or not
- progam to check the num is prime or not primec include main ( ) int numic clrscr ( ) scanf (dnum) ic while (iused for math function like sqrt ( ) and fabs ( ) main ( ) Float a b c xxdreal img printf (n Enter the values of abc in axbx c t) scanf (fff a b c) d b b a c if (d) d sqrt (fabs (d)) saqr used for square root fabs used for absolute value printf (n The roots are imaginary) real b (a) img d (a) printf (x f i f f real img) printf (x f i f f real img) else d sqrt (d) saqr used for square root fabs used for absolute value printf (n The roots are imaginary) real (b d) (a) img (b d) (a) printf (x f t x f f real img) getch ( ) Some Important Questions What is a control statement
- Explain with examples Write a program to check the given number is prime or not
- Write a program to check number is palindrome or not
- A program to read a string and check for palindrome void main ( ) char st int len i pal clrscr ( ) printf (Enter string of our choice) gets (st) len strlen (st) for (i i (len) i) if (sti

- stleni) pal if (pal) printf (n The input string is not palindrome) else printf (n the input string is palindrome) getch ( ) Some Important Questions What is an array

file2_topic_126: table, conversion, standard, input, charater

- The standard input functions are scanf ( ) getchas ( ) getch ( ) gets ( ) etc
- Similarly the output functions which are used to display the result on the screen are called standard output functions
- The standard library stdioh provides functions for input and output
- Types of IO The inputoutput functions are classified into two types i Formatted functions ii
- The input function scanf ( ) and output Program function printf ( ) fall under this category
- This is possible in C using the scanf function
- The input data can be entered into the computer from a standard input device keyboard by means of the C library function scanf
- Several of the more frequently used conversion characters are listed below Table Conversion Charater Table Example inlcude main ( ) char item int partno float cost scanf (s d f item partno cost) The following data items could be entered from the standard input device when the program is executed
- Example include main ( ) char line scanf (n line) A variation of this feature which is often more useful is to precede the characters within the square brackets by a circumflex (or caret)
- Several of the more frequently used conversion characters are listed below Table Conversion Charater Table Note l for long int h for signedunsigned short L for double
- For example printf ( ) scanf ( ) sqrt ( ) getch ( ) etc

file2_topic_127: formatted, unformatted, format, output, refers

- Unformatted functions Formatted Functions Formatted functions allow the input read from the keyboard or the output displayed on screen to be formatted according to our requirements
- While displaying a certain data on screen we can specify the number of digits after decimal point number of spaces before the data the position where the output is to be displayed etc using formatted functions

- Formatted Input Formatted input refers to an input data that has been arranged in a particular format
- Formatted Output Formatted output refers to the output of data that has been arranged in a particular format
- screen This function can be used to output any combination of numerical values single character and strings
- argn) where control string refers to a string that contains formatting information and arg arg argn are arguments that represent the individual output data item
- Format for Integer Output wd where w is the integer number specifying the minimum field width of output data
- The display is right justified For example char str MY NEPAL Unformatted Functions Unformatted functions do not allow the user to read or display data in desired format
- (PUBack) include include void main ( ) char class country printf (sclass) printf (cclass) printf (s rc class class) getch ( ) Write the various input unformatted function and describe any two
- Write the various output unformatted function and describe any two

file2_topic_128: cost, item, partno, scanf, arg

- scanf ( ) stands for scan formatted
- The general syntax of scanf function is scanf (control string arg arg argn) where control string refers to a string containing certain required formatting information so also known as format string and arg arg argn are arguments that represent the individual input data items
- Example include main ( ) int a b c scanf (d d d a b c) Suppose the input data items that are entered as Then the following assignment will result a b c If the data had been entered as Then the assignment would be a b c Now suppose that the data had been entered as Then the assignments would be a b c Finally suppose that the data had been entered as The resulting assignments would now be a b c Example include main ( ) int i float x char c If the data items are entered as T The output would now be The remaining two input characters (and T) will be ignored
- Example include main ( ) short ix iy long lx ly double dx dy scanf (hd ld lf ix ly dx) The control string in the first scanf function indicates that the first data item will be assigned to a short decimal integer variable
- The control string in the second scanf function indicates that the first data item will

have a maximum field width of characters and it will be assigned to short octal integer variable the second data item will have a maximum field width of characters and it will be assigned to a long hexadecimal integer variable and the third data item will have a maximum field width of characters and it will be assigned to double precision variable

- Example include main ( ) int partno float cost scanf (s d f item partno cost) If the corresponding data item are input fasterner fasterner is assigned to item and will be assigned to cost
- In contrast to scanf ( ) function the arguments in a printf ( ) function do not represent memory addresses and therefore are not preceded by ampersands
- For example include main ( ) char item int partno float cost printf (sdf item partno cost) Suppose fastener and have been assigned to name partno and cost
- It offers an alternative function of scanf ( ) function for reading strings
- Unlike scanf ( ) function it doesnot skip whitespaces

file2_topic_129: eof, returns, success, error, significance

- The function returns the number of data items that have been entered successfully
- When the program receives this signal the file reading function returns EOF which is a constant defined in the file stdioh and its value is
- It returns the integer written to file on success and EOF on error getw ( ) used to read integer value from a file
- It returns the next integer from the ip file on success and EOF on error
- What is the significance of EOF

file2_topic_130: argu, command, line, argument, argc

- Actually the arguments represent pointers that indicate the addresses of the data items within the computers memory
- In fact main can take two arguments called argi and argu and the information contained in the command line (ie
- The variable argc is an argument counter that the number of arguments on the command line
- The argu is an argument vector and represents an array of character pointers that point to the command line arguments
- For instance for the command line given above argc is three and argu is an array of

three pointers to strings as shown below argu PROGRAM argu X_FILE argu Y _FILE In order to access the command line arguments we must declare the main function and its parameters as follows main (argc argu) int argc char argu WAP that will receive a filename and a line of text as command line arguments and write the text to the file
- The command line F_ TEXT AAAAAA BBBBBB GGGGGG Each word in the command line is an argument to the main and therefore the total number of argument is
- The argument vector argv points to the string TEXT and therefore the statement
- The argument vector argv contains the entire command line in the memory and therefore the statement printf (sn is argvi) prints the argument from the memory

file2_topic_131: time, handle, heshe, cumbestsome, soon

- Thus the user may enter whatever he or she wishes and then presses the Enter Key
- In both cases input accepted as soon as the character typed
- At the same time it is cumbestsome and time consuming to handle large volume of data through keyboard
- It takes a lot of time to enter the entire data
- If the user makes a mistake while entering the data heshe has to start from the beginning again
- If the same data is to be entered again at some later stage again we have to enter the same data

file2_topic_132: width, field, item, specified, characters

- It is possible to limit the number of such characters by specifying a maximum field width for that data item
- To do so an unsigned integer indicating the field width is placed within the control spring between the The data item may contain fewer characters than the specified filed width
- However the number of characters in the actual data item can not exceed the specified field width
- Any characters that extend beyond the specified filed width will not be read
- Such leftover characters may be incorrectly interpreted as the components of the next data item
- The flags may be blank space or

- Flags Meanings Data item is left justified within the field
- The blank spaces required to fill the TableFlags ¾ Field Width Optional The field width is an integer specifying the minimum output field width
- If the number of characters in the corresponding data item is less than the specified field width then the data item will be preceded by enough leading blanks to fill the specified field
- If the number of characters in the data item exceeds the specified field width then additional space will be allocated to the data item so that the entire data item will be displayed
- If the length of the variable is less than the specified field width then the variable is right justified with leading blanks

file2_topic_133: parameters, parameter, regular, needed, specifier

- The arguments can be written as constants single variable or array names or more complex expressions
- Parameters (as many as needed) Each parameter consists of a data type specifier followed by an identifier like any regular variable declaration
- The different parameters are separated by commas
- These are known as actual parameter

file2_topic_134: control, mean, statements, different, statement

- There are two types of control statements
- What do you mean by control statements
- What is control statement
- Describe different control statement

file2_topic_135: year, ifelse, large, leap, yearn

- Decisions if ifelse nested ifelse switch Loops Loops are used when we want to execute a part of program or block of statement several times
- Following are decision making statements if statements ifelse statements else if statement Nested ifelse statement switch statement if statement The if statement is a

powerful decision making statement and is used to control the flow of execution of statements

- For example if (condition) if (condition) statementA else statement A Here we have ifelse inside both if block and else block else if (condition) statementB else statementB Program to find whether a year is leap or not include main ( ) int year clrscr ( ) printf (Enter year) if (year) if (year) printf (Leap year n) else printf (Not leap yearn) getch ( ) This can also be written in place of nested if else as if ((year year) year) printf (d is a leap yearn year) else printf (d is not a leap yearn year) Program to find largest number from three given number include main ( ) int a b c large clrscr ( ) printf (Enter three numbers) scanf (ddd a b c) if (ab) if (ac) large a else large c else if (bc) large b else large c printf (Largest number is dn large) getch ( ) Output Enter the numbers Largest num is else if statement This is a type of nesting in which there is an ifelse statement in every else part except the last else part
- Statements switch break continue goto Switch Statement This is a multidirectional conditional control statement
- The statements under case can be any valid C statements like ifelse while for or even another switch statement

file2_topic_136: factorial, transfer, fib, temp, num

- Print Enter a number whose factorial is to be calculated
- Initilize fact to and counter i to For inum factfacti i End of For Print fact as factorial of the number num
- WAP to find the factorial of a number using recursive method long int factorial (int n) if (n) return ( ) else return (nfactorial (n)) main ( ) int num printf (Enter a number) scanf (d num) printf (The factorial is ld factorial (num)) Output Enter a number The factorial is The Towers of Hanoi The Towers of Hanoi is a well known childrens game played with three poles and a number of different size disk
- The Tower of Hanoi_solved using recursion include void transfer (int n char from char to char temp) function prototype main ( ) left Center Right int n printf (Welcome to the Tower of Hanoi nn) printf (How many disk) scanf (d n) printf (n) transfer (n l R C) void transfer (int n char from char to char temp) transfer n disks from one pole to another n number of disks from origin to destination temp temporary storage if (n) move n disk from origin to temporary transfer (n from temp to) move nth disk from origin to destination printf (Move disk dfromctocn n from to) move n disks from

temporary to destination transfer (n temp to from) return Program Fibonacci number by recursion include include int fib (int x) if (x x) return else return (fib (x) fib (x)) main ( ) int i n no clrscr ( ) printf (how many no in series) for (i in i) no fib (i) printf (d no) getch ( ) WAP to add the natural numbers using recursive method long int add (int n) If (n) return else return (n add (n)) main ( ) int num clrscr ( ) printf (Enter How many numbers) scanf (d num) printf (The sum of natural number is ld add (num)) getch ( ) Output Enter How many numbers The sum of natural numbers is More about main ( ) We know that every C program should have one main ( ) function and that it marks the beginning of the program

- Write a program to calculate factorial of n number by using recursive function where n is the number inputted by user
- Write a program in C to allow a user to enter an integer number interactively and display its factorial value

file2_topic_137: condition, statement, stat, true, executed

- This statement is used to test a condition and take one of two possible actions If the condition is true then a single statement or a block of statements is executed (one part of the program) other wise another single statement or a block of statements is executed (other part of the program)
- syntax if (condition) if (condition) statement statement statement n There can be a single statement or a block of statements after the if part
- Flowchart Fig Flowchart of if control statement Here if the condition is true (nonzero) then statement is executed and if it is false (zero) then the next statement which is immediately after the if control statement is executed
- The syntax is if (condition) if (condition) statement else statement statement else statement Flowchart Fig Flowchart of ifelse control statement Here if the condition is true then statement is executed and if it is false then statement is executed
- After this the control transfers to the next statement which is immediately after the ifelse control statement
- if (condition) if (condition) statementA statementA else elseif (condition) if (condition) statementB statementB elseif (condition) else statementC if (condition) else statementC statement D else statement D The flowchart for else if statement is False True Condition False True Condition Stat A False True Condition Stat B Stat D Stat C Next Statement Program to find out the grade of a student when the marks of

subjects are given

file2_topic_138: nested, falling, ladder, frequently, block

- else if statement Nested if else statement We can have another if else statement in the if block or the else block
- This is called nested ifelse statement
- This type of nesting is frequently used in programs and is also known as else if ladder
- This is known as falling through cases

file2_topic_139: recordn, delete, insert, databasen, modify

- Create databasen) printf (
- Insert new recordn) printf (
- Modify a recordn) printf (
- Delete a recordn) printf (

file2_topic_140: extern, tda, specification, storage, class

- The keyword auto may be used storage class specification while declaration of variable
- A variable declared inside a function without storage class specification auto is by default an automatic variable
- An external variable must begin with the storage class specifier extern
- A variable declared outside a function without storage class specification extern is by default an external variable but defined after some function
- A program to illustrate the global variables int a void func ( ) a printf (tda) void main ( ) clrscr ( ) printf (tda) func ( ) printf (tda) Output Illustratration of extern variable main ( ) extern float marks func ( ) extern float marks float marks global space but defined after function The extern declaration does not allocate storage space variables
- The extern declaration of marks inside the function informs the compiler that marks is a float type extern variable defined somewhere else in the program

file2_topic_141: static, increment, register, variable, initial

- Default initial value of such type of variable is an unpredictable value which is often called garbage value
- Storage class refers to the performance of a variable and its scope within the program
- Initial Value This is the initial value assigned by the language to a variable if no value is assigned to variable explicitly by the programmer
- The default initial value for these variable is zero
- Static variables Static variables are declared by writing keyword static in front of the declaration
- static type var_name A static variable is initialized once and the value of a static variable is retained between function call
- If a static variable is not initialized then it is automatically initialized to O
- Program with auto variable Program with static variable void increment ( ) void increment ( ) int i static int i printf (dn i) printf (dn i) i i main ( ) main ( ) clrscr ( ) increment ( ) increment ( ) increment ( ) increment ( ) increment ( ) increment ( ) increment ( ) increment ( ) getch ( ) getch ( ) Output Register variable Register variables are special case of automatic variables
- If a variable is declared as register variable then it is stored in the CPU register
- The scope of register variables is local to the block in which they are declared
- Rules for initializations for register variables are the same as for automatic variables
- What is a static variable
- Write a programin C that uses a static variable
- Memory Block We have studied that it is necessary to declare a variable before using it since compiler has to reserve space for it
- The data type of the variable also has to be mentioned so that the compiler knows how much space need to be reserved
- For example int age The compiler reserves consecutive bytes from memory for this variable and associates the name age with it
- Like all other variables it also has a name to be declared and occupies some space in memory

file2_topic_142: scope, life, block, defined, local

- The scope of it is local to the block in which the variable is defined
- Again its life is till the control remains within the block in which the variable is defined
- Scope Scope of a variable can be defined as the region over which the variable is

visible or valid
- The scope is global ie
- Its scope is local to the block in which the variable is defined
- Again the life time is global ie

file2_topic_143: pole, disks, disk, poles, centre

- Each disk has a hole in the centre allowing it be stacked around any of the poles
- Initially the disks are stacked on the leftmost pole in the order of decreasing size ie
- the largest on the bottom and the smallest on the top
- The object of the game is to transfer the disks from the leftmost pole to the rightmost pole without ever placing a larger disk on the top of a smaller disk
- Only one disk may be moved at a time and each disk must always be placed around one of the poles
- The general strategy is to consider one of the poles to be the origin and another to be the destination
- The third pole will be used for immediate storage
- Thus allowing the disks to be moved without placing a larger disk over a smaller one
- Assume there are n disks numbered from smallest to largest
- If the disks are initially stacked on the left pole the problem of moving all n disks to the right pole can be stated in the following recursive manner
- Move the top n disks from the left pole to the centre pole
- Move the nth disk (the largest disk) to the right pole
- Move the n disks on the centre pole to the right pole
- In order to program this game we first label the poles so that the left pole is represented as L the centre pole as C and the right pole as R We then construct a recursive function called transfer that will transfer n disks from one pole to another
- Let us refer to the individual poles with the char type variable from to and temp for the origin destination and the temporary storage respectively
- Thus if we assign the character L to from R to and C to temp we will in effect be specifying the movement of n disks from the leftmost pole to righmost pole using the centre pole for immediate storage

file2_topic_144: source, content, file, welcome, college

- The for loop that follows immediately write the remaining arguments the file TEXT
- Program also prints two output one from the file TEXT and the other from the system memory
- include void main ( ) FILE fp fp fopen (Ctesttxt w) if (fp NULL) else printf (File has been successfully created) fputs (Welcome to Eastern College of Engineering fp) fclose (fp) output goto C drive and see the text file testtxt in notepad where the content is Welcome to Program to open the file texttxt created above read its content display to the screen include Void main ( ) FILE fp char s fp fopen (ctesttxt r) if (fp NULL) printf (In file can not be opened) exit ( ) fgets (s fp) printf (in the text from file is ts s) fclose (fp) getch ( ) OUTPUT The text from file is Welcome to Eastern College of Engineering
- Program that opens a file and copies all its content to another file
- Take source destination file from user include Void main ( ) FILE fsource fdest char ch source dest printf (In Enter source file name t) gets (source) printf (Enter destination file name t) gets (dest) fsource fopen (source r) if (fsource NULL) printf (n source file can not be opened) exit ( ) fdest fopen (dest w) if (fdest NULL) exit ( ) while ((ch fgetc (source)
- The syntax is long ftell (FILE fp) Some Programs Program to write some text welcome to my college to a data file in binary mode
- Read its content and display it include Void main ( ) FILE fptr char c fptr fopen (testtxt wb) if (fptr NULL) printf (n File can not be created) fputs (welcome to my college fptr) printf (The content from file n) while (cc fget c ffptr))
- EOF) printf (c C) fclose (fptr) output The Content from file Welcome to my college
- Read and display the same from the file
- Write a program that read a line of text and store file then print the content of file
- Write a program to open a file in write mode take input form the keyboard and write it to the file

file2_topic_145: acceptable, elements, illustrated, perfectly, element

- int a is a declaration of one dimensional array of type int
- Its elements can be illustrated as st element nd rd th th element a a a a a The elements of an integer array a are stored continuous memory locations
- int a b b a not acceptable if (ab) _ _ _ _ not acceptable we can assign integer values to these individual element directly a a a a a A loop can be used to input and output of

the elements of array
- a a a a a a int a In above example the values are less than the elements of array
- int a is perfectly acceptable

file2_topic_146: sizen, storage, follow, declaration, like

- declaration of an array data_type array_namesize or if we want to add storage classes then that look like storage_class data_type array_namesize where storage_class refers to the storage class of the array
- data_type is the data type of array
- Initialization of array The array is initialized like follow if we need time of declaration data_type array_namesize value value valuen For eg
- The general format for declaring multidimensional array is data_type array_name size size sizen We can add storage class in above declaration if necessary like follow storage_class data_type array_namesize size sizen where storage_class part is optional

file2_topic_147: accessing, index, multidimensional, array, help

- array_name is name of the array
- It is user defined name for array
- The name of array may be any valid identifier
- The accessing function for array is array_nameindex or subscript For eg
- Accessing elements of multidimensional array We can access the multidimensional array with the help of following accessing function array_name index index
- An array name can be named as an argument for the prototype declaration and in function header

file2_topic_148: stringh, manipulating, fprintf, fget, functions

- In C header file stringh provides special function for manipulating strings
- In order to use these function we must include stringh file
- Unformatted IO functions a) Character IO functions fget ( ) It is used to read a character from a file
- Formatted IO functions fprintf ( ) This function is used to write some integer float char

or string to a file

file2_topic_149: dma, freeing, allocating, dynamic, run

- include void main ( ) int a int i j clrsc ( ) for (c i i) for (jo j j) Printf (dt ((ai) j)) printf (n) getch ( ) output Dynamic Memory Allocation (DMA) The process of allocating and freeing memory at run time is ka Dynamic Memory Allocation
- In such situation DMA will be useful
- This is ka DMA
- DMA refers allocating and freeing memory at run time

file2_topic_150: realloc, allocated, previously, space, free

- This reserves the memory required by the program and returns this resource to the system once the use of reserved space utilized
- In some cases it is not possible to know the size of the memory required well ahead and to keep a lot of memory reserved is not also good practice
- There are library functions malloc ( ) calloc ( ) free ( ) and realloc ( ) for memory management
-) free ( ) The builtin function frees previously allocated space by calloc malloc or realloc function
- This can be done using free ( ) function
- Ths this function is used to release the space when it is not required
-) realloc ( ) This function is used to modify the size of previously allocated space
- Sometimes the previously allocated memory is not sufficient we nned additional space and sometime the allocated memory is much larger than necessary
- In both situations we can change the memory size already allocated with the help of function realloc ( )
- Program to illustrate the use of realloc ( ) include include include void main ( ) charname clrscr ( ) name (char) malloc ( ) strcpy (nameB
- The space is created like II

file2_topic_151: calloc, ptr, malloc, allocates, size

- Theses functions are defined within header file stdlibh and alloch) malloc ( ) It allocates requested size of bytes and returns a pointer to the first byte of the allocated space
- Its syntax is as ptr (data_typr malloc (size_of_block) where ptr is a pointer of type data_type
- The malloc ( ) returns a pointer to an area of memory with size size_of_block
-) calloc ( ) The function calloc ( ) provides access to the C memory heap which is available for dynamic allocation of variable_size block of memory
- Unlike malloc ( ) the function calloc ( ) accepts two arguments no_of_blocks and size_of_block
- This parameter no_of_blocks specifies the number of items to allocate and size_of_block specifies the size of each item
- The function calloc ( ) allocates multiple blocks of storage each of the same size and then sets all bytes to zero
- One important difference between malloc ( ) and calloc ( ) is that calloc ( ) initializes all bytes in the allocated block to zero
- Its syntax is ptr (data_typecalloc (no_of_blocksize_of_each_block) For example x (int) calloc (sizeof (int)) or x (int) calloc ( ) The above statement allocates contiguous space for blocks each of size bytes ie
- Its syntax is Free (ptr) Where ptr is a pointer to a memory block which has already been created by malloc ( ) calloc ( ) or realloc ( ) function
- Its syntax is as If the original allocation is done by the statement ptrmalloc (size) then reallocationof space may be done by the statement ptrrealloc (ptrnewsize) This function allocates a new memory space of new size to the pointer variable ptr and returns a pointer to the first byte of new memory block and on failure the function return NULL

file2_topic_152: var, semester, branch, structure, struct

- Defining a structure arrays of structures structures within structures The general syntax for declaration of a structure is storage_class struct name data_type member data_type member data_typen membern where the storage_class is optional
- For example) struct student char name int roll_no char branch int semester) static struct data int day char month_name int year Creating structure variable In above section we have studied about how we declare structure but have not created the

structure variable

- Method of creating structure variable I creating structure variable at the time of declaration Example struct student char name int roll_no char branch int semester var var var In this example three structure variables name var var var are created
- struct student structure declaration char name int roll_no char branch int semester var name roll_no branch semester var name roll_no branch semester name var roll_no branch semester struct student var var var Accessing member of structure The accessing concept of member is structure_variable_name
- mark is equivalent to (varmarks) Implies that the dot operator acts first and thenunary operatorFor example struct bio char name char address long phno struct bio b b for accessing bphno Initialization of structure A structure is initialized like other data type in C The values to be initialized must appear in order sa in the definition of structure within braces and separated by commas
- Its syntax is struct structure_name structure_variable value value value _ _ _ _ valuen ( ) struct particular int rn int age char sex We can initialize the structure at the time of variable creation like struct_particular per m By this is assigned to m of per is assigned to age of per and m is assigned to sex of per structure
- ( ) struct bio int age int rn char sex int phno If we write the following statement strwt bio b m In above example C compiler assigns following value to each member age sex m rn phno So C compiler will automatically initialized zero to those members who are not initialized
- For example Create a structure named date that has day month and year as its members

file2_topic_153: structure, pointer, members, member, accessed

- Note that the space in the memory for a structure is created only when the structure variable are defined
- C doesnot allow the initialization of individual structure member within its definition
- To store address of a structure type variable we can define a structure type pointer variable as normal way
- This declaration for a pointer to structure does not allocate any memory for a structure but allocates only for a pointer
- To use structures members through pointer p memory must be allocated for a structure by using function malloc ( ) or by adding declaration and assignment as given

below p b Here the base address of b can assign to p pointer
- An individual structure member can be accessed in terms of its corresponding pointer variable by writing ptr_variable member where is called arrow operator and there must be pointer to the structure on the left side of this operator
- C compiler first calculates the size of all members and then reserves the space which is highest among them
- How structure members are accessed using pointer

file2_topic_154: employee, salary, fptr, emp, birthday

- Use this structure to read and display employees name id dob and salary include void main ( ) struct date int day int month int year struct employee char name int id struct dat dob float salary emp printf (Name of Employee t) scanf (s empname) printf (n ID of employee t) scanf (d empid) printf (n Day of Birthday t) scanf (d empdobday) printf (n month of Birthday t) scanf (d empdobmonth) printf (n Year of Birthday t) scanf (d empdobyear) printf (salary of Employee t) scanf (d empsalary) printf (nn The Detail Information of Employee) printf (n Name t id t day t month t year t salary) printf (n n) printf (stdtdtdtdtfempnameempidempdobday empdobmonth empdobyear empdobsalary) output Name of Employee Teena Id of Employee Day of Birthday Month of Birthday Year of Birthday Salary of Employee The Detail Information of employee Name ID Day Month Year Salary Teena Processing a Structure (PU) WAP to read records of employee (Enter relevant fields Name address salary Id) void main ( ) struct employee char nam char Address float salary int ID int i j float temp struct emp clrsc ( ) printf (enteremployee Information) for (i i i) for (ji j j) if (empisalaryempjsalary) temp empisalary empisalary empjsalary empjsalary temp printf (Information of employees having highest salary In) printf (nNametttAddressttSalarytDtn) printf (n n) for (i i i) printf (sttsttftd empiName empiAddress empiSalary empiID) getch ( ) PU Create a user defined array structure student record having members physics chemistry and mathematics
- Use this structure to read the name age and salary of employee and write entered information to a file employeedat in D drive include void main ( ) struct employee char empName int age float salary struct employee emp FILE fptr ftpr fopen (demployeedat wb) if (fptr NULL) printf (File can not be entered) exit ( ) printf (Employee Name It) scanf (s emp
- empName) printf (employee age It) scanf (d empage) printf (salary of the

employee It) scarf (f empsalary) printf (In writing this information to a file _ _ _ In) fwrite (emp sizeof (emp) fptr) getch ( ) Create a structure named student that has name roll and marks as members

- wb) if (fptr NULL) printf (File cant be created) exit ( ) for (i i itt) printf (n Enter Information of student No d n i) printf (Name t) scanf (s siname) printf (n Rollt) scanf (d siroll) printf (n Marks t) scanf (f tempMarks) simarks tempMarks printf (n working Information to file _ _ _ _ _ n) fwrite (s size of (s) fptr) rewind (fptr) printf (n reading same content from file _ _ _ _ _ n) fread (st size of (st) fptr) printf (n student Name t Roll t Marks) printf (n _ n) for (i i i) printf (sttdtf n stiname stiroll stimarks) fclose (fptr) getch ( ) In Program read employee information again and again until user wants to add more employees

- Finally write a program to search information of a particular employee from the file

- include include struct employee char name int age float salary structemployee emp FILE fptr char yes_no name int dataFound clrscr ( ) fptr fopen (Cemployeetxt wb) if (fptr NULL) printf (File can not be created) exit ( ) Do printf (employee Name t) scanf (s empname) printf (employee age t) scanf (d empage) printf (salary of the employee t) scanf (f empsalary) fwrite (emp size of (emp) fptr printf (Do you want to add another employee

- Press Y or Y it) fflush (stdin) yes_no getchar ( ) while (yes_no y yes_no y) printf (Enter the name of employee which is to be searched) fflush (stdin) gets (name) rewind (fptr) while (fread (emp size of (emp) fptr)) if (strcmp (empname name)) dataFound printf (Name t Age t t Salarly n) printf (n _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ — _ _ _ _ _ _ _ _ n) print (stdtfempname empage empsalary) If (dataFound) print (n Matching data not found) getch ( ) Difference between text and binary mode Some Important Questions

file2_topic_155: students, structure, having, object, mathematics

- Write a program having a structure of student type
- Make use of array of structure to input information of students
- Write a program to read the name roll no and name of five students using array of structure object
- Create a user defined array structure student record having members physics chemistry and mathematics
- Use this structure to read and display records of students

file2_topic_156: files, data, necessary, binary, information

- These problems invite the concept of data file in which data can be stored on the disks and read whenever necessary without destroying data
- The data file allows us to store information permanently and to access and alter that information whenever necessary
- Mainly there are two types of data files
- The standard data files are again subdivided into text files and binary files
- The text files consist of consecutive characters and these characters can be interpreted as individual data item
- The binary files organize data into blocks containing contiguous bytes of information

file2_topic_157: acess, record, fortyforth, sequentially, random

- For example if we want to acess the fortyfourth record then firsat forty three records should be read sequentially to reach the fortyforth record
- In random acess data can be accessed and processed randomly ie
- in this case the fortyforth record can be accessed directly
- There is no need to read each record sequentially if we want to acess a particularly record
- Random acess takes takes less time than the sequential acess

file2_topic_158: text, graphics, mode, header, erases

- The text mode functions are concerned with placing text in certain area of screen
- Some text mode graphics function clrscr ( ) This function erases the text window
- In text mode we can address only location ( ) but in graphicsmode we can address individual pixel or dots on the screen
- If we want to use graphics mode functions then we have to add graphics h header file but for text mode function no need of graphics h header file

file2_topic_159: graphics, mode, adapter, monitor, driver

- They work with any graphics monitor and adapter

- The graphics mode function require a graphics monitor and adapter card such as CGA EGA or VGA
- This makes computer screen in specified graphics mode
- This function initializes the graphics system by loading a graphics driver from disk then putting the system into graphics mode
- This variable specifies the graphics driver are applicable only in graphics mode and they communicate directly with monitor
- Depending on what adapter is used one of these driver gets selected
- The some constants defined in graphics h file for this argument are DETECT ( ) CGA ( ) EGA ( ) VGA ( ) etc
- Closing Graphics Mode Once a program has finished its job using the graphics facilities then it should restore the system to the mode that was previously in use (ie
- the graphics mode should be closed)
- If graphics mode is not closed explicitly by the programmer undesirable effects may be felt
- If we want to use the graphics mode functions then our first job is to set our computer mode to graphics mode

file2_topic_160: gotoxy, joining, cursor, draws, lineral

- The general syntax Window (left top right bottom) gotoxy ( ) The gotoxy ( ) library function points the cursor position within a text window its syntax is gotoxy (xy) text color ( ) function It is used to give color of any text
- The syntax of line ( ) function is line (x y x y) where x y x y are integer type and they represent the coordinate (x y) and (x y)
- The above command draws a line joining two points with coordinates (x y) and (x y)
- lineral ( ) the function lineral (x y) draws a line joining the current cursor position and a point at a distance of x in the horizontal and y in vertical direction
- lineto ( ) The function lineto (x y) draws a line joining the current cursor position and a point with coordinates x and y

file2_topic_161: color, background, getpixel, setbkcolor, textbackground

- The syntax of text color ( ) is text color (color constant) textbackground ( ) function This function is used to set background color of each character

- The syntax is textbackground (color constant) Some color constants Color No
- Color name Black Blue Green Cyan Red Magenta Brown etc
- Its syntax is Putpixel (int x int y int color) getpixel ( ) gets color of specified pixel
- Its syntax is integer_variable getpixel (int x int y) setcolor ( ) It changes current drawing foreground color
- Its syntax is setcolor (int color) setbkcolor ( ) It changes the background color
- Its syntax is setbkcolor (int color) line ( ) The line ( ) function can draw a line
- Int P Drawpoly (P) setlinestyle ( ) This function can select different style of line

file2_topic_162: initgraph, mode, bgi, graphics, path

- Graphics Mode Graphic function Initialization In order to initialize graphics we initgraph ( ) function
- Its general syntax is initgraph (graphics_driver graphics_mode path to driver) where graphics_driver is a variable of type int initialized to some constant that defined in graphicsh
- The closegraph ( ) fuction is used to restore the screen to the mode it was in before we called initgraph ( ) and deallocates all memory allocated by the graphics system
- The syntax of closegraph ( ) sunction is closegraph ( ) Graphics Mode In text mode we are restricted to display text or graphics character but in graphics mode we can display points lines and compleax shapes
- Function initgraph ( ) sets our mode for graphics work
- The syntax of initgraph ( ) function is initgraph (driver mode path of bgi files) where driver and mode both are integer type and path for bgi files means where our BGI files are in disk
- If we want to set the mode of computer is graphics mode by initgraph ( ) function then we have to write the following set of statements intdriver mode driverDETECT initgraph (driver mode Ctcbgi) we are assuming that our bgi are in sub directory Ctcbgi therefore the path of bgi files is written in initgraph ( ) function like Ctcbgi The statement driverDETECT will check our hardware and select appropriate values for argument to function initgraph ( ) Some Graphics Mode Graphic Functions putpixel ( ) Plot a point with specified color

file2_topic_163: arc, draws, circle, rectangle, ellipse

- rectangle ( ) The function rectangle (x y x y) draw rectangle where point (x y) is left top corner point of rectangle and point (x y) is right bottom corner
- ( ) y x ( ) y x circle ( ) The function circle (x y r) draws a circle of radius r The coordinates of the centre of the circle is (x y)
- r ( ) y x arc ( ) Function arc (x y a a r) draws an arc on the screen starting from angle a to a
- The radius of the circle of which the arc forms a part is r and x y are its centre coordinates
- For example arc ( ) Above statement draws an arc like ellipse ( ) The function ellipse (x y c c a a x y) draws an ellipse of centre (c c)
- The a and a are start and end angle of the arc x and y are the xaxis and yaxis radii

file2_topic_164: vertices, polygon, fillpoly, drawpoly, closed

- a drawpoly ( ) Function drawpoly (int n int p) draws n vertices of polygon
- To draw a closed polugon with n vertices we must pass n coordinates
- fillpoly ( ) It draws and fills polygon
- Its syntax is fillpoly (int n int p) To draw a closed polygon with vertices we must pass n coordinates to fillpoly ( )

file2_topic_165: style, thickness, pattern, respectively

- Its syntax is setlinestyle (int style patern thickness) The type of style and thickness is int type and the type of pattern is unsigned int type
- Where style are SOLID_LINE DOTTED_LINE CENTRE_LINE DASHED_LINE USERBIT_LINE or integer number respectively
- The pattern is required only if user defined style (USERBIT_LINE) is used
- The thickness parameter can have value NORM_WIDTH or THICK_WIDTH or integer value or respectively

file2_topic_166: circle, draw, rectangle, detect, ctcbgi

- Simple program using built in graphical function PU Program in c to draw a line a circle a rectangle and an ellipse include main ( ) int gd gm clrscr ( ) gd DETECT initgraph (gd

gm Ctcbgi) setcolor ( ) line ( ) circle ( ) rectangle ( ) ellipse ( ) getch ( ) closegraph ( ) PU Program to draw concentric circle having radius m and units include main ( ) int gd DETECT gm clrscr ( ) initgraph (gd gm Ctcbgi) setcolor ( ) circle ( ) circle ( ) circle ( ) getch ( ) closegraphc ( ) Program to draw an arc include main ( ) int gd DETECT gm clrscr ( ) initgraph (gd gm Ctcbgi) arc ( ) getch ( ) closegraph ( ) Program to draw a polygon include main ( ) int gd DETECT gm int P ( ) clrscr ( ) initgraph (gd gm Ctcbgi) drawpoly (P) fillpoly (P) getch ( ) closegraph ( ) Some Important Questions Write a program to draw (PU) (i) Line (ii) Circle (iii) Rectangle (iv) Arc

- Write a program using graphics to draw a rectangle and a circle
- (PU BACK) Write a program in C to draw a circle a rectangle and an ellipse
- (PU) Write a program in C that can display a circle and a rectangle
- Choose centre and radius of the circle as well as coordinates of the rectangle on your own
- (PU) Write a program to draw concentric circles having radius and units