

merged_topic_0: structure, member, structures, connects, initializing / structure, struct, member, book, bio

- A structure definition forms a template that may be used to create structure variable
- The general format of a structure definition is as follows `struct tag_name data_type member data_type member` Array VS Structure Both the array and structures are classified as structured data types as they provides mechanism that enables us to access and manipulate data in relatively easy manner
- Structure can have elements of different types
- But in the case of a structure first we have to design and declare a data structure before the variable of that type are declared and used
- Rules of initializing structure There are few rules to keep in mind while initializing structure variable at compiletime
- Note Two variables of the same structure type can be copied the same ways as ordinary variables
- And C programming doesn't permit any logical operations on structure variables
- The same is true of pointers used with structures
- Defining pointer to Structure `Struct sturcture_name pointer_variable` Example `Struct complex ss` `ss` assign the address of the structure variable `s` to `s` Accessing member of structure using pointer variable `Pointer_variable member_namevalue` Example `sr` Note The
- operator connects a structure with a member of the structure the operator connects a pointer with a member of the structure
- For example in analyzing the marks obtained by a class of students we may use a templates to describe student name and marks obtained in various subjects and then declare all the students as structure variables
- In such cases we may declare an array of structures each element of the array representing a structure variable
- We can also use tag name to define inner structures
- +++++
- 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_type member` where the `storage_class` is optional
- The `struct` is a keyword
- Arun Kumar Yadav Lecturer Eastern College of Engineering Biratnagar Computer

Programming in C 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 then unary operator For 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 as in the definition of structure within braces and separated by commas

- C does not allow the initialization of individual structure member within its definition

- Its syntax struct structure_name structure_variable value value value _ _ _ _ value n () 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 m 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 Downloaded from www.jayaram.com np Downloaded from www.jayaram.com np If we write the following statement struct bio b m In above example C compiler assigns following value to each member age sex m rn phno So C compiler will automatically initialize zero to those members who are not initialized

- For example Struct Book char name float price Struct Book b Programming In C Let Us C Programming with C Structure within structure (Nested Structure) One structure can be nested within another structure in C In other words the individual members of a structure can be other structure as well

- To store address of a structure type variable we can define a structure type pointer variable as normal way

- Let us consider a structure struct book char name float price Then we can define structure variable and pointer variable of structure type book b b is structure variable book p p is pointer variable of structure type Where b is simple variable of structure type book where as p is pointer type variable which points or can store address of structure book type variable

- 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

- How structure members are accessed using pointer
- typedef struct FILE A file pointer is a pointer to a structure of type FILE

merged_topic_1: label, jump, goto, forward, branch / goto, label, statement, labeled, unconditionally

- Example include include void main() int abcl printf(nEnter two numbers) 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

- 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 label statement goto label Forward jump Backward jump Compiled by Sudip Lama 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

- ++++

- Arun Kumar Yadav Lecturer Eastern College of Engineering Biratnagar Computer Programming in C 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

- 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_2: formatted, putchar, getchar, char, input / getche, putch, character, getchar, puts

- For example char name namegetchar() The getchar () function accept any character keyed in

- 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 printf(nThe name is s name) getch() Compiled by Sudip Lama Formatted input Formatted
```

input refers to an input data that has been arranged in particular format

- Following are the specifications for reading character strings ws or wc Compiled by Sudip Lama 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

- For example char name 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

- +++++

- Thus the user may enter whatever he or she wishes and then presses the Enter Key

- The functons getchar() putchse() gets() puts() getch() getche() putch() are considered as unformatted functions

- getchar() and putchar () The getchar() function reads a character from a standard input device

- The general syntax 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 getche() 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 conio.h and hence we should include this in our program using the instruction include <conio.h>
- Arun Kumar Yadav Lecturer Eastern College of Engineering Biratnagar Computer Programming in C 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
- In both cases input accepted as soon as the character 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()
- In order to use these function we must include string.h file
- At the same time it is cumbersome 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

merged_topic_3: structure, member, structures, connects, initializing / var, creating, structure, semester, branch

- A structure definition forms a template that may be used to create structure variable
- The general format of a structure definition is as follows `struct tag_name data_type member data_type member` Array VS Structure Both the array and structures are classified as structured data types as they provides mechanism that enables us to access and manipulate data in relatively easy manner
- Structure can have elements of different types
- But in the case of a structure first we have to design and declare a data structure before the variable of that type are declared and used
- Rules of initializing structure There are few rules to keep in mind while initializing structure variable at compiletime
- Note Two variables of the same structure type can be copied the same ways as ordinary variables
- And C programming doesnt permit any logical operations on structure variables
- The same is true of pointers used with structures
- Defining pointer to Structure `Struct sturcture_name pointer_variable` Example `Struct complex ss` `ss` assign the address of the structure variable `s` to `s` Accessing member of structure using pointer variable `Pointer_variable member_namevalue` Example `sr` Note The
- operator connects a structure with a member of the structure the operator connects a pointer with a member of the structure
- For example in analyzing the marks obtained by a class of students we may use a templates to describe student name and marks obtained in various subjects and then declare all the students as structure variables
- In such cases we may declare an array of structures each element of the array representing a structure variable
- We can also use tag name to define inner structures
- ++++
- 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 Downloaded from www.jayaramcomnp Downloaded from www.jayaramcomnp 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

- For example struct employee char name int empid float salary emp Where emp is an array of employee structures

- For example Create a structure named date that has day month and year as its members

merged_topic_4: distance, marks, student, struct, total / grade, marks, singh, students, average

- of a student and marks obtained by him in subjects Declare 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 sitotal for(jjj) int roll int marks scanf(dsimarksj) int total sitotal sitotal simarksj void input(struct student s) int ij for(iii) printf(nEnter the roll number of student) scanf(dsiroll) printf(nEnter marks for subjects) Compiled By Sudip Lama void display(struct student s) int ilocation max maxstotal location for(iii) if(maxsitotal) serching highest marks location maxsitotal locationi printf(nRecord of student who score highest marks) printf(nRoll number of student dslocationroll) printf(nEnter marks for subjects) for(iii) printf(nMarks in d subject dislocationmarks) 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

- include include struct distance int feet int inch Compiled By Sudip Lama 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 d struct distance d) struct distance d dfeetdfeetdfeet dinchdinchdinch dfeetdinch dinchdinch return d struct distance sub(struct distance d struct 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 subtraction of time is) display(t) getch() Compiled By Sudip Lama Nested Structure Structure within a structure means nesting of 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 (ffffm m m m) total mmmm per total if(per) grade A elseif(per) grade B elseif(per) grade C elseif(per) grade D 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) grade D Er

- M Singh) printf(n Names name) name(char)realloc(ame) strcpy(name,captain B M Singh) printf(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) printf(n Enter marks of each studentsn) p(float)malloc(nsizeof(float)) for(i in i) Er

- Arun Kumar Yadav Lecturer Eastern College of Engineering Biratnagar Computer Programming in C 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) 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() return() Downloaded from wwwjayaramcomnp Downloaded from wwwjayaramcomnp 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

- Feed the marks obtained by three students in each subjects and calculate the total marks of each student void main() struct student int physics int chemistry int mathematics int total struct student std int i printf(Enter the marks in physics chemistry and mathematics of student) for(i i i) printf(Marks of students d i) scanf(ddd stdiphsics stdichemistry stdimathematics) Downloaded from wwwjayaramcomnp Downloaded from wwwjayaramcomnp stditotal stdiphsics stdichemistry stdimathematics printf(Information of students) printf(n n) printf(studenttttotal marks t) printf(n n) fori i i) printf(nStudentdttdn i sditotal) Structures and pointers Pointers can be used also with structure

merged_topic_5: precedence, operator, levels, associativity, decides /

precedence, associativity, operators, arithmetic, left

- The precedence and associativity of and operators are the same as those of unary and unary
- 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
- There are distinct levels of precedence and an operator may belong to one of these levels
- The operator at the higher level of precedence is evaluated first
- Precedence is defined as the rules that decides the order in which different operator are applied
- The operator of the same precedence are evaluated either from left to right or from right to left depending on the level
- This is known as the associativity property of an operator
- Associativity is also defined as the rule that decides the order in which multiple occurrences of the same level operator are applied
- Compiled by Sudip Lama The following table provides a complete list of operators their precedence levels and their rules of association
- The following is the rule of Hierarchy arithmetic operators in FORTRAN
- +++++
- Precedence Associativity Precedence of an operator in C is defined as the order of evaluation
- Operations with a high precedence are Er
- Table Precedence of Arithmetic Operators

Precedence	Arithmetic Operators	Descriptions
multiplication	division	modular division
addition	subtraction	Associativity

- The order in which consecutive operations within the same precedence group are carried out is known as associativity
- 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 associativity
- The dot operator has precedence and associativity is left to right

merged_topic_6: content, mode, file, existing, opens / mode, new, exist, existing,

erased

- File opening Modes The different types of file opening modes are Compiled By Sudip Lama
- 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 cannot modify existing contents
- +++++
- 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_7: prototype, declaration, local, function, global / prototype, func, int, definition, void

- 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 include double convert (int) function prototype Er
- Function Declaration or Prototype The function declaration or prototype is model or blueprint of the function
- 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
- These functions can be written as void func (int int) main() func(a b) void func(int c int d) statements Er
- For eg include void swap(int int) function prototype void main() int x y clrscr() x swap (xy) function call by value printf (x d n x) Er
- A function is called from the definition of

merged_topic_8: distance, marks, student, struct, total / employee, student, salary, roll, information

- of a student and marks obtained by him in subjects Declare 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 sitotal for(jjj) int roll int marks scanf(dsimarksj) int total sitotal sitotal simarksj void input(struct student s) int ij for(iii) printf(nEnter the roll number of student) scanf(dsiroll) printf(nEnter marks for subjects) Compiled By Sudip Lama void display(struct student s) int ilocationmax maxstotal location for(iii) if(maxsitotal) serching

highest marks location maxx total location i printf(n Record of student who score highest marks) printf(n Roll number of student d s location roll) printf(n Enter marks for subjects) for(iii) printf(n Marks in d subject d s location marks i) 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 mark 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 feet inches using structure

- include include struct distance int feet int inch Compiled By Sudip Lama void input(struct distance d) printf(n Enter the feet and inches) scanf(dddfeetdinch) void display(struct distance d) printf(n Feet d t Inched t dfeetdinch) struct distance add(struct distance d struct distance d) struct distance d dfeetdfeetdfeet dinchdinchdinch dfeetdinch dinchdinch return d struct distance sub(struct distance d struct distance d) struct distance d dinchdinchdinch dfeetdfeetdfeet if(dinch) dfeetdfeet dinchdinch return d void main() struct distance ttt printf(n Enter the starting distance in feet and inches) input(t) printf(n Enter the stoping distance in feet and inches) input(t) tadd(tt) printf(n addition of two time is) display(t) tsub(tt) printf(n The subtraction of time is) display(t) getch() Compiled By Sudip Lama Nested Structure Structure within a structure means nesting of structure

- +++++

- 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 roll t) 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 entr marks Er

- 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) Downloaded from wwwjayaramcomnp Downloaded from wwwjayaramcomnp 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 Roll Er

- 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[50]; int id; struct date dob; float salary;} emp; printf("Name of Employee "); scanf("%s", emp.name); printf("ID of employee "); scanf("%d", &emp.id); printf("Day of Birthday "); scanf("%d", &emp.dob.day); printf("Month of Birthday "); scanf("%d", &emp.dob.month); printf("Year of Birthday "); scanf("%d", &emp.dob.year); printf("Salary of Employee "); scanf("%f", &emp.salary); printf("\n\n The Detail Information of Employee\n"); printf("Name\tid\tday\tmonth\tyear\tsalary\n"); printf("%s\t%d\t%d\t%d\t%d\t%f", emp.name, emp.id, emp.dob.day, emp.dob.month, emp.dob.year, emp.salary); output: Name of Employee: Teena ID of Employee: 10 Day of Birthday: 10 Month of Birthday: 10 Year of Birthday: 2000 Salary of Employee: 10000 The Detail Information of employee: Name ID Day Month Year Salary Teena Processing a Structure WAP to read records of employee (Enter relevant fields: Name, address, salary, ID). Display the records of the top three employees which have the highest salary. Er

- Arun Kumar Yadav, Lecturer, Eastern College of Engineering, Biratnagar. Computer Programming in C. void main() {struct employee {char name[50]; char Address[50]; float salary; int ID;} emp; int i, j; float temp; struct emp; clrscr(); printf("Enter employee Information "); for(i=1; i<=5; i++) for(j=1; j<=5; j++) if(emp[i].salary > emp[j].salary) temp = emp[i].salary; emp[i].salary = emp[j].salary; emp[j].salary = temp; printf("Information of employees having highest salary In\n"); printf("Name\tAddress\tSalary\tID\n"); for(i=1; i<=5; i++) printf("%s\t%s\t%f\t%d", emp[i].name, emp[i].Address, emp[i].Salary, emp[i].ID); getch()} PU Create a user defined array structure student record having members physics, chemistry and mathematics.

- 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;} st; printf("Enter roll and marks\n"); scanf("%d%f", &st.roll, &st.marks); printf("Roll: %d, Marks: %f", st.roll, st.marks);}

- void main() {union student {int roll; float marks;} st; printf("Enter roll and marks\n"); scanf("%d%f", &st.roll, &st.marks); printf("Roll: %d, Marks: %f", st.roll, st.marks);}

- EOF) fputc(ch, fdest); printf("Successfully copied"); fclose(fsource); fclose(fdest); getch()} Program to understand fprintf() include: #include <stdio.h> struct student {char name[50]; 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 fopen(studentdat
- 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 fptr 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) scanf(f
empsalary) printf(In writing this information to a file _ _ _ In) fwrite(emp sizeof(emp) fptr)
getch( ) Downloaded from wwwjayaramcomnp Downloaded from wwwjayaramcomnp 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(stdtf 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
- Arun Kumar Yadav Lecturer Eastern College of Engineering Biratnagar Computer
Programming in C struct employee char name int age float salary struct employee 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( ) Downloaded from
wwwjayaramcomnp Downloaded from wwwjayaramcomnp Difference between text and
binary mode Text mode Binary mode

```

merged_topic_9: distance, marks, student, struct, total / student, students,

structure, roll, having

- of a student and marks obtained by him in subjects
- Declare 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 sitotal for(jjj) int roll int marks scanf(dsimarksj) int total sitotal sitotal simarksj void input(struct student s) int ij for(iii) printf(nEnter the roll number of student) scanf(dsiroll) printf(nEnter marks for subjects) Compiled By Sudip Lama 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 student dslocationroll) printf(nEnter marks for subjects) for(iii) printf(nMarks in d subject dislocationmarksj) 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
- include include struct distance int feet int inch Compiled By Sudip Lama 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 d struct distance d) struct distance d dfeetdfeetdfeet dinchdinchdinch dfeetdinch dinchdinch return d struct distance sub(struct distance d struct 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 subtraction of time is) display(t) getch() Compiled By Sudip Lama Nested Structure Structure within a structure means nesting of structure
- +++++
- Create a structure named student that has name roll marks and remarks as members
- 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
 - Write an array of structure to a file then read its content to display to the screen
- ```
Void main()
struct student {Char name; int roll; float marks; }
struct student s; int i; float tempMarks;
FILE fptr;
fptr = fopen("student.txt", "w");
```

merged\_topic\_10: format, printing, specification, integer, width / format, formatted, str, gender, unformatted

- 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
  - Inputting Integer Numbers The field specification for reading an integer number is %w%d The percentage sign(%) indicates that a conversion specification follows
  - %w is an integer number that specifies the field width of the number to read and %d known as data type character indicates that the number to be read is in integer mode
  - Format specifications that define the output format for display of each item
  - Commonly used printf format codes
- | Code | Meaning                                                                          |
|------|----------------------------------------------------------------------------------|
| c    | printing a single character                                                      |
| d    | printing a decimal integer                                                       |
| e    | printing a floating point value in exponent form                                 |
| f    | printing a floating point value without exponent                                 |
| g    | printing a floating point value either %e type or %f type depending on its value |
| i    | printing a signed decimal integer                                                |
| o    | printing an octal integer without leading zero                                   |
| s    | print an unsigned decimal integer                                                |
| x    | print a hexadecimal integer without leading 0x                                   |
- The following letters may be used as prefix for certain conversion characters
- %h for short integer %l for long integer or %double %L for long double
- Commonly used output format flags
- | Flag | Meaning                                   |
|------|-------------------------------------------|
| l    | Output is left justified within the field |
- 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(s s s ) r where n is the statement number s s s are the format specifications r Rules
  - I format The symbol %I is used to denote the integer quantities
  - The general I format specification is: %Iw Where w is the width of the integer data
  - Example This data can be describe by the format statement: FORMAT(I) 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

- The general form of the F format is F wd Where w is the total width of the number d is the decimal width c Temperature conversion REAL FC WRITE() FORMAT(X Enter the temperatureX THE VALUE) READ()C FORMAT(F) FC WRITE()F FORMAT(XTemperaturef) Unformatted Input and output c Temperature conversion REAL FC PRINT Enter the temperature READC FC PRINT TemperatureF Control Statement Unconditional GOTO statement This statement is used to transfer the control to any other statement unconditionally

- +++++

- 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 argn are arguments that represent the individual output data item

- Format for Integer Output Er

- int n Format Output printf (d n) printf (d n) wl 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 )

- The display is right justified For example char str MY NEPAL Format Output printf (s str) 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) M Y N E P A L Unformatted Functions Unformatted functions do not allow the user to read or display data in desired format

- Downloaded from wwwjayaramcomnp Downloaded from wwwjayaramcomnp 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 function used to clear console(display) screen clrscr( ) is pronounced clearscreen getch( ) function is used to hold the console screen

- (PUBack) b Find the output of the following program

- (PUBack) include include void main( ) char class country printf(sclass) printf(cclass) printf(src class class ) getch( ) Write the various input unformatted function and describe any two

- Write the various output unformatted function and describe any two

merged\_topic\_11: structure, member, structures, connects, initializing / employee, structure, named, salary, create

- A structure definition forms a template that may be used to create structure variable

- The general format of a structure definition is as follows struct tag\_name data\_type member data\_type member Array VS Structure Both the array and structures are classified as structured data types as they provide mechanism that enables us to access and manipulate data in relatively easy manner

- Structure can have elements of different types

- But in the case of a structure first we have to design and declare a data structure before the variable of that type are declared and used

- Rules of initializing structure There are few rules to keep in mind while initializing structure variable at compiletime

- Note Two variables of the same structure type can be copied the same ways as ordinary variables

- And C programming doesn't permit any logical operations on structure variables

- The same is true of pointers used with structures

- Defining pointer to Structure Struct sturcture\_name pointer\_variable Example Struct complex ss ss assign the address of the structure variable s to s Accessing member of structure using pointer variable Pointer\_variable member\_namevalue Example sr Note The

- operator connects a structure with a member of the structure the operator connects a pointer with a member of the structure

- For example in analyzing the marks obtained by a class of students we may use a templates to describe student name and marks obtained in various subjects and then declare all the students as structure variables

- In such cases we may declare an array of structures each element of the array representing a

structure variable

- We can also use tag name to define inner structures

- +++++

- struct student structure declaration char name int roll\_no char branch int semester structure variable creation Er

- Each element of the array emp will contain the structure of the type employee

- The another way to declare structure is struct employee char name int empid float salary struct employee emp Create a structure named student that has name roll marks and remarks as members

- Include this structure as a member in another structure named employee which has name id salary as other members

- Create a structure named employee having numbers empName age and salary

merged\_topic\_12: statement, switch, condition, block, expression / switch, case, break, 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

- Compiled by Sudip Lama 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 Test Expression The ifelse statement The general form is if (test expression) Trueblock statement(s) 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 true block or falseblock will be executed not both

- Entry True False Expression Falseblock Trueblock statement statement Statement x Compiled by Sudip Lama Nesting of ifelse statement The general form is if (test condition) if(test condition) statement statement 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 defaultstatement statementx The switch statement The switch statement tests the value of 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 Compiled by Sudip Lama The expression is an integer expression or characters

- Each these values should be unique within a switch statement

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

- k Arithmetic IF statement This statement is used to transfer the control depending upon the value of an expression whether negative zero or positive

- 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 Compiled By Sudip Lama 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

- This is called Nested usage of IFTHENELSE

- ++++

- The criteria for making the decision should be indicated clearly within the decision box

- 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 Downloaded from wwwjayaramcomnp Downloaded from wwwjayaramcomnp The keywords are all lowercase

- 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

- Statements switch break continue goto Switch Statement This is a multidirectional conditional control statement

- Sometimes there is a need in program to make choice among number of alternatives

- 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

- The statements under case can be any valid C statements like ifelse while for or even another switch statement

- 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

- Downloaded from [www.jayaram.com](http://www.jayaram.com) Downloaded from [www.jayaram.com](http://www.jayaram.com) Flowchart switch case constant Body of case constant (expression constant) case constant Body of case constant (expression 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 printf (firstn) Er

- Arun Kumar Yadav Lecturer Eastern College of Engineering Biratnagar Computer Programming in C 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

- Break statement Break statement is used inside lops and switch statements

- general syntax) break If a break statement is encountered inside a switch then all the statements following break are not executed and the control jumps out of the switch

- Program to understand the switch with break statement include main( ) int choice clrscr( ) printf (Enter your choice ) scanf (d choice) switch (choice) Downloaded from [www.jayaram.com](http://www.jayaram.com) Downloaded from [www.jayaram.com](http://www.jayaram.com) case print (Firstn) break break statement case printf(Secondn) break case printf (Thirdn) break default printf (Wrong choicen) end of switch getch( ) End of main( ) Output Enter your choice Program to perform arithmetic calculation on integers include main( ) char op int a b clrscr( ) printf (Enter a number operators and another num ) scanf (dcd a op b) switch (op) case printf (Result dn ab) break case printf (Result dn ab) case printf (Result dn ab) case printf (Result dn ab) default Er

- Describe the application of break and continue in Cprogramming

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

merged\_topic\_13: string, stringstring, char, printf, include / string, strcpy, strrev, reversed, accepts

- include void main() include void add(int nint r) int listresultij printf(nEnter the number ) int

ij for(iii) for(iii) for(jjj) scanf(dlistij) ri add(listresult) for(jjj) printf(The result is) for(iii) rinij  
printf(ndresulti) getch() Compiled By Sudip Lama 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 stringi include for(istringii) include void main() if(stringi ) char string int ic printf(n  
The number of word in clrscr() sentencedc) printf(n Enter the first string) getch() gets(string)  
strlen(string) Compiled By Sudip Lama 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

- The pointer string now points to the first character of the string Sagar String Example include  
include void main() char stringstring clrscr() printf(nEnter the string) gets(string) stringstring  
while(string) printf(nc is stored at address ustringstring) string printf(nThe length of string  
isdstringstring) getch() Write a program that reads two different strings

- 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 strlen(name) Compiled By Sudip Lama  
rjname! 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() Compiled By Sudip Lama  
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

- ++++

- char text Kathmandu printf(Number of characters dsizeof(text)) will generate the following  
output Number of characters Conditional Operators The operator

- Arun Kumar Yadav Lecturer Eastern College of Engineering Biratnagar Computer  
Programming in C strlen( ) This function returns an integer which denotes the length of string  
passed

- Its syntax is integer\_variable strlen(string) Program to find out the length of a string include  
include include void main( ) char name int len clrscr printf (Enter your name t) gets(name) len  
strlen(name) printf (n The number of character in your name is td len) getch( ) Output Enter  
Your Name ARUN KUMAR YADAV The number of character in your name is strcpy( ) The  
strcpy( ) function copies one string to another

- The function accepts two string as parameters and copies the second string character by character into the first one upto including the null character of the second string
- The syntax is `strcpy(destination_string source_string)` ie
- `strcpy(s s)` means the content of s is copied to s
- Yadav s `strcpy (sname) printf (The copied string is st s)` Downloaded from [www.jayaram.com](http://www.jayaram.com)np  
Downloaded from [www.jayaram.com](http://www.jayaram.com)np Output The copied string is Arun Kr
- Yadav `strcat( )` This function concatenates two strings ie
- it appends one string at the end of another
- This function accepts two strings as parameters and stores the contents of the second string at the end of the first
- Its syntax is `strcat (string string)` ie
- This function is useful for constructing and searching strings as arranged into a dictionary
- This function accepts two string as parameters and returns an integer whose value is ) less than if the first string is less than the second ) equal to if both are same ) greater than if the first string is greater than the second Two strings are compared character by character until there is a mismatch or end of one of strings is reached
- Its syntax `integer_variable strcmp(string string)` Program to illustrate the use of `strcmp( )`  
include Er
- `Strrev( )` This function is used to reverse all characters in a string except null character at the end of string
- Its syntax is `strrev(string)` For example `strrev(s)` means it reverses the characters in string s and stores reversed string in s Program to illustrate `strrev( )` include `main( )` `char name Manju Shree)` `char name strcpy (name name) printf (The reversed string of original string s is s name name)` Output The reversed string of original string Manju Shree is eerhS ujnaM

merged\_topic\_14: distance, marks, student, struct, total / students, roll, percentage, obtained, subjects

- of a student and marks obtained by him in subjects Declare 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 sitotal for(jjj) int roll int marks scanf(dsimarksj) int total sitotal sitotal simarksj void input(struct student s) int ij for(iii) printf(nEnter the roll number of student) scanf(dsiroll) printf(nEnter marks for subjects) Compiled By Sudip Lama void display(struct student s) int ilocationmax maxstotal location for(iii) if(maxsitotal) serching`

highest marks location maxx total location i printf(nRecord of student who score highest marks) printf(nRoll number of student d s location roll) printf(nEnter marks for subjects) for(iii) printf(nMarks in d subject d s location marks i) 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 mark 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 feet inches using structure

- include include struct distance int feet int inch Compiled By Sudip Lama void input(struct distance d) printf(nEnter the feet and inches) scanf(dddfeetdinch) void display(struct distance d) printf(nFeet d t Inched t dfeetdinch) struct distance add(struct distance d struct distance d) struct distance d dfeetdfeetdfeet dinchdinchdinch dfeetdinch dinchdinch return d struct distance sub(struct distance d struct 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 subtraction of time is) display(t) getch() Compiled By Sudip Lama Nested Structure Structure within a structure means nesting of structure

- +++++

- Problem There are students in a class who appeared in their final examination

- Their mark sheets have been given to you

- 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

- START READ INPUT ADD MARKS OF ALL SUBJECTS GIVING TOTAL PERCENTAGE TOTAL WRITE OUTPUT DATA Fig Flowchart Coding In order to make a program in any programming language what we have written is known as code

- 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
- Write a program in C that allows a user enter name age sex address number of songs recorded duet singers using a file

merged\_topic\_15: calling, pass, original, function, called / pass, reference, parameter, passed, mechanism

- 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 cannot be changed accidentally
- 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
- Therefore any changes introduced to the array elements are truly reflected in the original array in the calling function

- ++++

- 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
- 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

merged\_topic\_16: string, stringstring, char, printf, include / rajesh, string, ram, diff, greater

- include void main() include void add(int nint r) int listresultij printf(nEnter the number ) int ij for(iii) for(iii) for(jjj) scanf(dlistij) ri add(listresult) for(jjj) printf(The result is) for(iii) rinij printf(ndresulti) getch() Compiled By Sudip Lama 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 stringi include for(istringii) include void main() if(stringi ) char string int ic printf(n The number of word in clrscr() sentencedc) printf(n Enter the first string) getch() gets(string) strlen(string) Compiled By Sudip Lama 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

- The pointer string now points to the first character of the string Sagar String Example include include void main() char stringstring clrscr() printf(nEnter the string) gets(string) stringstring while(string) printf(nc is stored at address ustringstring) string printf(nThe length of string isdstringstring) getch() Write a program that reads two different strings

- 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 strlen(name) Compiled By Sudip Lama rjname! 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() Compiled By Sudip Lama 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

- ++++

- Initializing string A string is initialized as char name A R U N Then string name is initialized to ARUN

- But C offers special way to initialize the string as char name Arun The characters of the string are enclosed within a pair of double quotes

- Program to illustrate string initialization include include void main( ) char name ARUN KUMAR clrscr( ) printf (Your name is sn name) getch( ) Output Your name is ARUN KUMAR Arrays of strings Arrays of string means two dimensional array of characters

- Program to copy one string to another indlue include main( ) char name Arun Kr

- string string string Program to concate two strings include include main( ) char fname Arun

lname Kumar strcat (fname lname) printf (n The full name is s frame) Output The full name is Arun Kumar strcmp( ) This function compares two strings to find out whether they are same or different

- Whenever two characters in two strings differ the string which has the character with a higher ASCII value is greater
- For example consider two string ram and rajesh
- The first two character are same but third character in string ram and that is in rajesh are different
- Since ASCII value of character m in string ram is greater than that of j in string rajesh the string ram is greater than rajesh
- Arun Kumar Yadav Lecturer Eastern College of Engineering Biratnagar Computer Programming in C include main( ) char name name int diff printf (n Enter first string t) gets(name) printf (n Enter second string t) gets(name) diff strcmp(name name) if (diff) printf (s is greater than s by value d name name diff) print (s is same as s name name) Output Enter first string ram Enter second string rajesh ram is greater than rajesh by value
- The reverse of string abc is cba

merged\_topic\_17: passed, passing, array, argument, manipulating / pass, reference, parameter, passed, mechanism

- Pass by pointers method is often used when manipulating arrays and strings
- Compiled By Sudip Lama 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
- 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
- However this does not apply when an individual element is passed on argument
- The function must be called by passing only the array name
- +++++
- 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

- 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

merged\_topic\_18: format, printing, specification, integer, width / displayed, item, constant, character, conversion

- 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

- Inputting Integer Numbers The field specification for reading an integer number is w s d The percentage sign(%) indicates that a conversion specification follows

- w is an integer number that specifies the field width of the number to read and d known as data type character indicates that the number to be read is in integer mode

- Format specifications that define the output format for display of each item

- Commonly used printf format codes Code Meaning c printing a single character d printing a decimal integer e printing a floating point value in exponent form f printing a floating point value without exponent g printing a floating point value either etype or ftype depending on i printing a signed decimal integer o printing an octal integer without leading zero s print an unsigned decimal integer x print a hexadecimal integer without leading 0x The following letters may be used as prefix for certain conversion characters

- h for short integer l for long integer or double L for long double Commonly used output format flags Flag Meaning Output is leftjustified within the field

- 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(s s s ) r where n is the statement number s s s are the format specifications r Rules

- I format The symbol I is used to denote the integer quantities

- The general I format specification is Compiled By Sudip Lama 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

- The general form of the F format is F wd Where w is the total width of the number d is the

decimal width c Temperature conversion REAL FC WRITE() FORMAT(X Enter the temperatureX THE VALUE) READ()C FORMAT(F) FC WRITE()F FORMAT(XTemperaturef) Unformatted Input and output c Temperature conversion REAL FC PRINT Enter the temperature READC FC PRINT TemperatureF Control Statement Unconditional GOTO statement This statement is used to transfer the control to any other statement unconditionally

- +++++

- Several character constants are shown below Character constants have integer values that are determined by the computer's 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

- Several character constants and their corresponding values as defined by ASCII character set are shown below Downloaded from www.jayaram.comnp Downloaded from www.jayaram.comnp 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 single character 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

- Several of the more frequently used conversion characters are listed below Conversion Characters Meanings c Data item is a single character d Data item is a decimal integer e Data item is a floating point value f Data item is a floating point value g Data item is a floating point value Er

- These other characters are simply transferred directly to the output device where they are displayed

- Several of the more frequently used conversion characters are listed below Downloaded from www.jayaram.comnp Downloaded from www.jayaram.comnp Conversion Character Meaning c Data item is displayed as a single data d Data item is displayed as floating point e value

with an exponent `em` is displayed as a signed decimal `f` Data item is displayed as floating point without an exponent `g` Data item is displayed as floating point value using either `etype` or `ftype` conversion `i` Data item is displayed as a signed decimal integer `o` Data item is displayed as an octal integer `w` Data item is displayed as a string without a leading zero `u` Data item is displayed as an unsigned decimal integer Data item is displayed as a hexadecimal integer without the leading `x`

- Table Conversion Character Table Note `l` for long int `h` for signed unsigned short `L` for double

merged\_topic\_19: file, opening, data, defined, filename mode / file, eof, position, fseek, syntax

- 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 Student.txt Employ.dat 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`  
`fopen(filename, mode)` 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`

- This pointer which contains all the information about the file is subsequently used as a communication link between the system and the program

- +++++

- For example a linefeed (LF) which is Downloaded from [www.jayaram.com](http://www.jayaram.com) Downloaded from [www.jayaram.com](http://www.jayaram.com) referred to as a newline in C can be represented as `\n`

- The function returns the number of data items that have been entered successfully

- Its syntax is `fclose(ptr_variable)` On closing the file all the buffers associated with it are flushed and can be available for other files

- Unformatted IO functions a) Character IO functions `fgetc()` It is used to read a character from a

file

- Its syntax is `char_variable fgetc(file_ptr_variable) fputc( )` It is used to write a character to a file
- Its syntax is `fputc(char_variable file_ptr variable)` 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_variable control string list variables) fscanf( )` This function is used to read some integer float char or string from a file
- Its syntax is `fscanf(file_ptr_variable control string list_variables)` Creating data files Some sample programs create a file named `test.txt` and write some text Welcome to Eastern College of Engineering to the file
- name `stdmarks)` `fclose(fp)` End of File (EOF) The file reading function need to know the end of file so that they can stop reading
- When the end of file is reached the opening system sends an end of file signal to the program
- When the program receives this signal the file reading function returns EOF which is a constant defined in the file `stdio.h` and its value is
- File Pointer Device `Stdin` Standard input device (keyboard) `Stdout` Standard OUTPUT devices (screen) `Stderr` Standard error OUTPUT device (screen) Some other unformatted function Integer IO `getw( ) putw( )` for binary mode Record IO `fread( ) fwrite` `Putw( )` used to write integer value to file
- 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 file on success and EOF on error
- Their syntaxes are `int putw(int value FILE fp) int getw(FILE fp) fwrite( )` used for writing an entire block to a given file
- `fread( )` is used to read an entire block from a given file
- Their syntax are `fwrite(ptr size_of_array_or_structure number_of_array_or_structure fp)` `fread(ptr size_of_array_or_structure number_of_structure_or_array fp)` Random Access to File We can access the data stored in the file in two ways sequentially or random
- C supports these functions for random access file processing `fseek( ) ftell( ) rewind( ) fseek( )` This function is used for setting the file position pointer at the specified byte
- 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 Position SEEK\_SET Beginning of file SEEK\_CURRENT Current position SEEK\_END End of File rewind( ) This function is used to move the file position pointer to the beginning of the file syntax rewind(File fp) using rewind(fp) is equivalent to fseek(fp, 0, 0) ftell( ) This function returns the current position of the file position pointer
- The value is counted from the beginning of the file
- What is the significance of EOF

merged\_topic\_20: character, hold, permits, char, declared / california, null, text, extra, meaningless

- A string is defined as a character array that is terminated by null character
- For this reason the character arrays are declared one character longer than the largest string they can hold
- The general form of declaration of a string variable is char string\_name[size] Example char name like numeric array character arrays may be initialized when they are declared
- C permits a character array to be initialized in either of the following two forms char name[SAGARMATHA] C also permits us to initialize a character array without specifying the number of elements
- The following code declares an array of strings each of which can hold maximum valid character
- char name Notice that the second index has been given value ie
- An array of strings appears in memory as shown below
- If the name is a character variable which will hold the name of length at most character then must be declared as follows CHARACTER name CHARACTER AB then A and B are both character variables to hold data of length at the most each
- +++++
- 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 10th element will represent the null character ( ) which is 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
- The terminating null character is important because it is the only way the string handling functions can know where the string ends

merged\_topic\_21: appear, userdefined, declares, declaration / semicolon, executable, declaration, variables, declares

- This section also declares all the userdefined functions
- The declaration part declares all the variables used in the executable part
- Userdefined functions are generally placed immediately after the main function although they may appear in any order
- define statements may appear anywhere in the program but before it is referenced in 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
- A function may be used by many other program
- ++++
- 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 doesn't require semicolon
- It may be auto static extern and register It is optional

merged\_topic\_22: character, hold, permits, char, declared / array, dimensional, multidimensional, size, element

- A string is defined as a character array that is terminated by null character
- For this reason the character arrays are declared one character longer than the largest string they can hold

- The general form of declaration of a string variable is `char string_name[size]` Example `char name` like numeric array character arrays may be initialize when they are declared
- C permits a character array to be initialized in either of the following two forms `char name[SAGARMATHA]` C also permits us to initialize a character array without specifying the number of elements
- The following code declares an array of strings each of which can hold maximum valid character
- `char name` Notice that the second index has been given value ie
- A array of string appears in memory as shown below
- If the name is a character variable which will hold the name of length at most character then must be declared as follows `CHARACTER name` `CHARACTER AB` then A and B are both character variable to hold data of length at the most each
- `++++`
- 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
- Square brackets enclosing the size specification for text
- The declaration could also have been written `char text California` where size of the array is explicitly specified
- The array is another kind of variable that is used extensively in C An array is an identifier that refers to collection of data items that have the same name
- The individual data items are represented by their corresponding array element (ie
- the first data item is represented by the first array element etc)
- The individual array elements distinguished from one another by the value that is assigned to a subscript
- The size of this array will be equal to the value of `argc`
- Downloaded from [www.jayaram.com](http://www.jayaram.com) Downloaded from [www.jayaram.com](http://www.jayaram.com) Arrays and Strings Introduction An array is a collection of same type of data item which are stored in consecutive memory locations under a common name
- In array system an array represents multiple data items but they share same name
- The individual elements are characterized by array name followed by one or more subscripts or indices enclosed in square brackets
- `auto register static or extern`) Single and Multidimension arrays One or single dimensional

array There are several forms of an array in C one 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 from 0 to n-1 where n is the size of the array
- The general syntax of array is ie
  - declaration of an array `data_type array_name[size]` or if we want to add storage classes then that look like `storage_class data_type array_name[size]` where `storage_class` refers to the storage class of the array
  - `data_type` is the data type of array
  - Arun Kumar Yadav Lecturer Eastern College of Engineering Biratnagar Computer Programming in C `array_name` is name of the array
  - It is user defined name for array
  - The name of array may be any valid identifier
  - size of the array is the number of elements in the array
  - The size is mentioned within
  - For example `define size a size` The size of the array must be specified ie
    - should not be blank) except in array initialization
  - Initialization of array The array is initialized like follow if we need time of declaration `data_type array_name[size] value value value` For eg
    - In this case the value to each element is assigned like following `element element element element element` ie
      - missing element will be set to zero
- Accessing elements of array If we have created an array the next thing is how we can access (read or write) the individual elements of an array
  - The accessing function for array is `array_name[index]` or subscript For eg
  - 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
  - Downloaded from [www.jayaram.com.np](http://www.jayaram.com.np) Downloaded from [www.jayaram.com.np](http://www.jayaram.com.np) A one dimensional array of two dimensional arrays is called three dimensional arrays etc
  - The two dimensional array are very useful for matrix operations
  - 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
- The general format for declaring multidimensional array is `data_type array_name size size` We can add storage class in above declaration if necessary like `follow storage_class data_type array_namesize size` where `storage_class` part is optional
- 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
- It is important to remember that while initializing a dimensional array it is necessary to mention the second (column) size where as first size (row) is optional
- 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
- For example `char str` The first dimension (size) tells how many strings are in the array
- In two dimensional array the array point to element
- In three dimensional array the array name point to element etc
- In two dimensional array the second element is
- The programmer must know the size of the array or data in advanced while writing the program
- Since an array name is actually a pointer to the first element within the array it is possible to define the array as a pointer variable to represent an array requires some type of initial memory assignment before the array elements are processed

**merged\_topic\_23:** pointer, subtract, integers, increment, discussed / address, price, points, variable, pointer

- 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
- +++++

- 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 variables using address operator
- Program to print address of variable using include main( ) int sn float price printf(value of sn Address of sn\n sn) printf(value of price Address of price\n 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
- Arun Kumar Yadav Lecturer Eastern College of Engineering Biratnagar Computer Programming in C 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
- iptr sn Pointer are also variables so compiler will reserve space for them and they will also have some address
- If pointers are declared 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 same
- PP Now both pointer variable P and P contains the address of variable sn and points the same variable sn Downloaded from www.jayaram.comnp Downloaded from www.jayaram.comnp We can also assign constant zero to a pointer of any type
- 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
- What does p p and p represents if p is declared as integer pointer

merged\_topic\_24: character, hold, permits, char, declared / rajesh, string, ram, diff, greater

- A string is defined as a character array that is terminated by null character
- For this reason the character arrays are declared one character longer than the largest string they can hold
- The general form of declaration of a string variable is char string\_name[size] Example char name like numeric array character arrays may be initialize when they are declared
- C permits a character array to be initialized in either of the following two forms char name[SAGARMATHA] C also permits us to initialize a character array without specifying the number of elements
- The following code declares an array of strings each of which can hold maximum valid

character

- char name Notice that the second index has been given value ie
- A array of string appears in memory as shown below
- If the name is a character variable which will hold the name of length at most character then must be declared as follows CHARACTER name CHARACTER AB then A and B are both character variable to hold data of length at the most each
- +++++
- Initializing string A string is initialized as char name A R U N Then string name is initialized to ARUN
- But C offers special way to initialize the string as char name Arun The characters of the string are enclosed within a pair of double quotes
- Program to illustrate string initialization include include void main( ) char name ARUN KUMAR clrscr( ) printf (Your name is %s name) getch( ) Output Your name is ARUN KUMAR
- Arrays of strings Arrays of string means two dimensional array of characters
- Program to copy one string to another include include main( ) char name Arun Kr
- string string Program to concatenate two strings include include main( ) char fname Arun lname Kumar strcat (fname lname) printf (%s The full name is %s frame) Output The full name is Arun Kumar strcmp( ) This function compares two strings to find out whether they are same or different
- Whenever two characters in two strings differ the string which has the character with a higher ASCII value is greater
- For example consider two string ram and rajesh
- The first two character are same but third character in string ram and that is in rajesh are different
- Since ASCII value of character m in string ram is greater than that of j in string rajesh the string ram is greater than rajesh
- Arun Kumar Yadav Lecturer Eastern College of Engineering Biratnagar Computer Programming in C include main( ) char name name int diff printf (%s Enter first string t) gets(name) printf (%s Enter second string t) gets(name) diff strcmp(name name) if (diff) printf (%s is greater than %s by value %d name name diff) printf (%s is same as %s name name) Output Enter first string ram Enter second string rajesh ram is greater than rajesh by value
- The reverse of string abc is cba

merged\_topic\_25: pointer, tells, book, asterisk, slide / address, price, points,

## variable, pointer

- Compiled By Sudip Lama 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 cannot be added
- +++++
- 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 variables using address operator
- Program to print address of variable using include `main( )` `int sn` `float price` `printf(value of sn` `Address of sn` `sn` `sn)` `printf(value of price` `Address of price` `un 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
- Arun Kumar Yadav Lecturer Eastern College of Engineering Biratnagar Computer Programming in C `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`
- `iptr` `sn` Pointer are also variables so compiler will reserve space for them and they will also have some address
- 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 same
- PP Now both pointer variable `P` and `P` contains the address of variable `sn` and points the same variable `sn` Downloaded from [www.jayaram.com.np](http://www.jayaram.com.np) Downloaded from [www.jayaram.com.np](http://www.jayaram.com.np) We can also assign constant zero to a pointer of any type
- 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`
- What does `p` `p` and `p` represents if `p` is declared as integer pointer

merged\_topic\_26: comma, left, right, expression, simple / assignment, expression, identifier, expressions, assignments

- 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
- Conditional Expression RIGHT TO LEFT Assignment Operator RIGHT TO LEFT Comma Operator LEFT TO RIGHT Compiled by Sudip Lama Managing input and output operation Reading a character Reading a single character can be done by using the function getchar
- The expression may be any simple variable or structure variable or an expression using simple variables
- +++++
- (PU) Downloaded from [www.jayaram.com.np](http://www.jayaram.com.np) Downloaded from [www.jayaram.com.np](http://www.jayaram.com.np) Operators Expressions Individual constants variables array elements and function references can be joined together by various operators to form expression
- (i(f)) false Assignment Operators There are several different assignment operators in C All of them are used to form assignment expressions which assign the value of an expression to an identifier
- The most commonly used assignment operator is
- Assignment expressions that make use of the operator are written in the form identifier expression where identifier generally represents a variable and expression represents a constant a variable or more complex expression
- Downloaded from [www.jayaram.com.np](http://www.jayaram.com.np) Downloaded from [www.jayaram.com.np](http://www.jayaram.com.np) Here are some typical assignment expressions that make use of the operator
- The assignment operator is used to assign a value to an identifier where as the equality operator is used to determine if two expressions have the same value
- Assignment expressions are often referred as assignment statements
- Multiple assignments of the form are permissible as identifier identifier identifier n In such situation the assignments are carried out from right to left
- Multiple assignment identifier identifier expression
- is equivalent to identifier (identifier expression) and so on with right to left resting for additional multiple assignments
- They are also known as shorthand assignment operators
- expression expression is equivalent to expression expression expression Expression



Equivalent Expression a b a ab a b a ab a b a ab a b a ab Er

- Arun Kumar Yadav Lecturer Eastern College of Engineering Biratnagar Computer Programming in C The general form of shorthand assignment operators variable operator variable (or expression) The use of shorthand assignment operator has three advantages
  - For example Downloaded from wwwjayaramcomnp Downloaded from wwwjayaramcomnp value (x y x y) Here is assigned to x and is assigned to y and so expression xy is evaluated as () ie
  - In C any nonzero value is regarded as true while zero is regarded as false
  - Thus the following assignments are valid
  - dot operator) Downloaded from wwwjayaramcomnp Downloaded from wwwjayaramcomnp
- Passing Structure to functions Like any other variable a structure variable can also be passed to a function

merged\_topic\_27: file, opening, data, defined, filenamemode / opened, file, opening, mode, fopen

- 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
- This pointer which contains all the information about the file is subsequently used as a communication link between the system and the program
- +++++
- The file name should end with the characters c like program c labc etc
- Then the command under Ms DOS operating system would load the program stored in the file

program c ie

- The linking is done by the command LINK programobj which generates the executable code with the filename programexe
- While using these library functions the entire data is lost when either the program is terminated or the computer is turned off
- Opening a file established a link between the program and the operating system
- 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
- Whenever a file is opened a structure of type FILE is associated with it and a file pointer that points to this structure identifies this file
- The function fopen( ) is used to open a file
- For example FILE fp fp fp fopen (myfile.txt w) fp fopen (yourfile.dat r) The file opening mode specifies the way in which a file should be opened (ie
- In other word it specifies the purpose of opening a file
- 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
- After closing the file connection between file and program is broken
- Explain in brief the various file opening modes
- Explain in brief the steps involved in opening a file

merged\_topic\_28: pointer, subtract, integers, increment, discussed / operators, equality, greater, operator, shift

- 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 not 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 relational 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
- Arun Kumar Yadav Lecturer Eastern College of Engineering Biratnagar Computer Programming in C In addition to the relational and equality operators C also includes the unary operator
- 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
- Logical negation Ones complement Pointer reference Address sizeof Sizeof operator Operators Description Associativity Precedence Rank Multiplication Left Right Division Modulus Addition Left Right Subtraction Right Downloaded from [www.jayaram.com.np](http://www.jayaram.com.np) Downloaded from [www.jayaram.com.np](http://www.jayaram.com.np) Rightshift Right Greater than Greater than or equal to Equality Left Right
- What are the various types of operators used in C language

merged\_topic\_29: index, base, address, nchanged, arrayindex / array, dimensional, multidimensional, size, element

- In c arrays index numbering starts with
- Example include include void main() int xbx int yay int z clrscr() printf(nEnter the value) printf(nEnter the value) printf(nThe value of z isdz) printf(nOriginal value of b is ub) printf(nChanged value of b is ub) printf(nOriginal value of a is ua) aa printf(nChanged value of a is ua) Compiled By Sudip Lama 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) 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 Compiled By Sudip Lama Pointer to Array in Function
- As an example to explain that a program where each element of array will be added by a constant
- +++++
- 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
- Square brackets enclosing the size specification for text
- Te declaration could also have been written `char text California` where size of the array is explicitly specified
- The array is another kind of variable that is used extensively in C An array is an identifier that refers to collection of data items that have the same name
- The individual data items are represented by their corresponding array element (ie
- the first data item is represented by the first array element etc)
- The individual array elements distinguished from one another by the value that is assigned to a subscript
- The size of this array will be equal to the value of `argc`
- Downloaded from [www.jayaram.com.np](http://www.jayaram.com.np) Downloaded from [www.jayaram.com.np](http://www.jayaram.com.np) Arrays and Strings Introduction An array is a collection of same type of data item which are stored in consecutive memory locations under a common name
- In array system an array represents multiple data items but they share same name
- The individual elements are characterized by array name followed by one or more subscripts or indices enclosed in square brackets
- 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 from 0 to n-1 where n is the size of the array

- The general syntax of array is `data_type array_name[size]` or if we want to add storage classes then that look like `storage_class data_type array_name[size]` where `storage_class` refers to the storage class of the array
- `data_type` is the data type of array
- Arun Kumar Yadav Lecturer Eastern College of Engineering Biratnagar Computer Programming in C `array_name` is name of the array
- It is user defined name for array
- The name of array may be any valid identifier
- size of the array is the number of elements in the array
- The size is mentioned within
- For example `int a[10];` The size of the array must be specified ie 10
- should not be blank) except in array initialization
- Initialization of array The array is initialized like follow if we need time of declaration `data_type array_name[size] value value value` For eg `int a[10] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};`
- In this case the value to each element is assigned like following element element element element element ie
- missing element will be set to zero Accessing elements of array If we have created an array the next thing is how we can access (read or write) the individual elements of an array
- The accessing function for array is `array_name[index]` or subscript For eg `a[0]`
- 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
- Downloaded from [www.jayaram.com.np](http://www.jayaram.com.np) Downloaded from [www.jayaram.com.np](http://www.jayaram.com.np) A one dimensional array of two dimensional arrays is called three dimensional arrays etc
- The two dimensional array are very useful for matrix operations
- 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 `data_type array_name[size1][size2]`
- 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

- The general format for declaring multidimensional array is `data_type array_name size size`. We can add storage class in above declaration if necessary like `follow storage_class data_type array_name size size` where `storage_class` part is optional.
- 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.
- It is important to remember that while initializing a dimensional array it is necessary to mention the second (column) size where as first size (row) is optional.
- 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.
- For example `char str`. The first dimension (size) tells how many strings are in the array.
- In two dimensional array the array point to element.
- In three dimensional array the array name point to element etc.
- In two dimensional array the second element is.
- The programmer must know the size of the array or data in advanced while writing the program.
- Since an array name is actually a pointer to the first element within the array it is possible to define the array as a pointer variable to represent an array requires some type of initial memory assignment before the array elements are processed.

## file1\_topic\_30: section, lama, sudip, compiled, object

- Compiled by Sudip Lama Computer Software. The software is what really gives the computer life.
- Compiled by Sudip Lama Compiler. A compiler is a program that translates program (called source code) written in some high level language into object code.
- Compiled by Sudip Lama Object code. Object code is the code produced by a compiler.
- Compiled by Sudip Lama Important Features of Algorithm.
- No object code is stored and there is no compilation.
- Compiled by Sudip Lama 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 final step in producing and Compiled by Sudip Lama Basic Structure of C programming. Documentation Section Link Section Definition Section Global Declaration Section `main()` Function Section Declaration part Executable part Subprogram section Function Funct ion

(Userdefined functions) Function n The documentation section consists of a set of comment lines giving the name of the program provides instructions to the compiler to link function from the system library

- Compiled by Sudip Lama Token Token is the smallest unit of a program ie
- Compiled by Sudip Lama
- They are Compiled by Sudip Lama
- Compiled by Sudip Lama Practice Question
- vvv Compiled By Sudip Lama

file1\_topic\_31: checks, computer, operating, programmer, antivirus

- It is a set of instructions that tells the computer what to do when the computer operator does something
- Solaris runs on Sun SPARC machines
- 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
- 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 facilitates topdown modular programming

file1\_topic\_32: software, application, solving, applications, needs

- Software can actually be placed into two separate categories system of applications
- Application software Application software is the software that is developed for a particular use
- Most of these process our data
- There are thousands of application software
- 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\_33: software, disk, operating, environment, handle

- 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 are 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
- These software run over the environment provided by the system software
- When this software is installed and you are connected to an ISP (Internet Service Provider) you have access to worldwide information

file1\_topic\_34: examples, word, browsers, excel, quattro

- Word processing software allows you to process and manipulate text
- Examples are Corel WordPerfect and Microsoft Word
- Examples are Quattro Pro and Excel
- Internet browsers give us access to the world
- Common examples are Netscape and Explorer

file1\_topic\_35: data, console, large, volumes, oriented

- 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 data 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 a 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\_36: languages, level, high, lowlevel, programming

- Programming languages are mainly categorized into two types low level and high languages
- Compiled by Sudip Lama 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 high level 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 1950s (low level or high level languages)

### file1\_topic\_37: machine, language, assembly, programs, understood

- These can be divided into two types machine language and assembly language
- Machine Language Machine language is the lowest level programming language which is the only language understood by a computer
- While easily understood by computers machine languages are almost impossible for humans to use because they consist entirely of numbers
- Early computers were usually programmed using machine language

- Assembly Language As the machine language involves only numbers specific to a machine developers thought that if they can represent those numbers by some symbols it would be easier to make programs
- So they developed assembly languages in which those numbers are replaced by some symbols (called mnemonic)
- As machine language is the only language understood by computers they also developed a program that could convert the programs written in symbols into machine language
- Like the machine language assembly language programs are also dependent on the machine because numbers may vary from machine to machine
- But the flexibility it offered was it could be made to run on other machines by slightly changing the code or changing the assembler
- Moreover its easier to learn assembly language than the machine language
- Like assembly language programs programs written in a highlevel language also need to be translated into machine language
- To execute the program however the programmer must translate it into machine language
- When we purchase programs we usually receive them in their machinelanguage format
- For these languages to work on the computer it must be translated into machine language

#### file1\_topic\_38: languages, cobol, interpreted, basic, algol

- Now there are dozens of different languages such as Ada Algol BASIC COBOL C C FORTRAN LISP Pascal and Prolog etc
- 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
- The most frequently used interpreted language is BASIC

#### file1\_topic\_39: interpreter, interpreters, programs, compilers, executes

- 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
- Compilers require some time before an executable program emerges
- 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
- Source code must go through several steps before it becomes an executable program
- There are two kinds of translators compilers and interpreters

#### file1\_topic\_40: languages, code, interpreted, object, compilers

- Because compilers translate source code into object code which is unique for each type of computer many compilers are available for the same language
- Unlike the compiled languages
- 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\_41: section, brace, closing, parts, executable

- This means that we can execute them directly but cannot read or modify them
- Every C program must have one main() function section
- This section contains two parts declaration part and 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
- All section except the main function section may be absent when they are not required
- This is a nonexecutable statement

#### file1\_topic\_42: flowchart, algorithms, solve, algorithm, flowcharts

- 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
- 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 Compiled by Sudip Lama For examples Read numbers form user and display the resulting sum
- start Read a b cab Display Flowchart to find the largest among three entered numbers Start input A B C Is AB Is AC Is BC Y N N Y Print A Print C Print B Compiled by Sudip Lama Coding Writing the program is called coding
- Write an algorithm and flowchart to generate the following

file1\_topic\_43: requirements, processing, specifying, inputs, input

- Specifying the input requirements
- Specifying the processing requirements
- Compiled by Sudip Lama 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
- This document should contain statements on the programs objectives output specifications input requirements processing requirements and feasibility
- Quantities that have specified relation to inputs

file1\_topic\_44: algorithm, inputs, outputs, ensure, initially

- (cid) Ensure that the algorithm has single entry and exit point
- Inputs Any algorithm can have zero or more inputs
- Inputs may be given initially or as the algorithm runs
- Outputs Any algorithm may result in one or more outputs

#### file1\_topic\_45: immediately, termination, execution, forcing, resumes

- Finiteness Every algorithm should lead to a session of task that terminates after a finite number of steps
- Translation and execution occur immediately one after another one statement at a time
- On exit the program continues with the statement immediately after the body of the loop
- Execution resumes at the statement immediately following the body of the terminated statement
- Instead of forcing termination it force the next iteration of the loop to take place statement
- In a recursive function there must be a reachable condition for its termination otherwise the function will be invoked endlessly
- If the condition is true the statement is executed and then goes to the next statement

#### file1\_topic\_46: hcf, rev, include, int, void

- For example Read numbers form user and display the resulting sum
- Read numbers and solve in variables say A and B
- include include void main() int abl printf(nEnter two numbers) labab printf(nThe largest number is d l) getch()
- Write a program to find the HCF for any two numbe entered by user
- include include int hcf(int fint s) if(fs) return s return (sfs) void main() int ab clrscr() printf(nEnter the number) printf(The Hcf is d hcf(ab)) getch() Compiled By Sudip Lama
- 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) 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) getch() Note

#### file1\_topic\_47: cid, able, conditions, programmer, crashing

- Here are some of the qualities of a good program (cid) It should be easily readable and

understandable by people other than the original programmer

- (cid) It should be efficient increasing the programmers productivity
- (cid) It should be reliable able to work under all reasonable conditions and always get the correct output
- (cid) 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
- (cid) It should be easy to maintain and support after installation

file1\_topic\_48: keyword, lowercase, colon, underscore, parenthesis

- Keyword All keyword have fixed meaning and these meaning cannot 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)
- Cannot 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 format specifications s s must be enclosed within parenthesis
- The specification s sr must be separated by commas
- Compiled By Sudip Lama GO TO In the GO TO statement the number referenced cannot be a variable example GO TO I is not valid
- The subscript must be given within parenthesis after the variable name

file1\_topic\_49: character, digits, characters, alphabets, succeeding

- These are userdefined names and consist of a sequence of letters and digits with a letter as a first character
- 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
- This statement displays the character contained in the variable\_name at the terminal
- for character and string
- A string of character is called a character constant

- 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
- A variable which acquires only character string is called a character variable

#### file1\_topic\_50: 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 cannot be used as variable names

#### file1\_topic\_51: conversion, type, typename, higher, automatically

- 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

#### file1\_topic\_52: getchar, interactively, turn, waits, pressed

- 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
- This could create problem when we use getchar () in a loop interactively

### file1\_topic\_53: output, newlines, blanks, tabs, unwanted

- This mean when we enter single character input the newline character is waiting in the input queue after getchar () return
- 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
- So the first character of the output must be made a blank space so that the loss does not affect the output

### file1\_topic\_54: scanf, notation, field, inputting, reads

- It can be performed using scanf function
- The general format of scanf is scanf (control string argargarnn) 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 is 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
- For example scanf(f f f x y z) Inputting Character Strings Scnf can input strings containing more than one character
- 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



file1\_topic\_55: 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\_56: positive, symbol, assumed, negative, sign

- or will precede the signed numeric item
- (with o or x) causing octal and hex item to be preceded by and x
- 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 can have sign( or )
- If there is no sign the exponent is assumed to be positive
- The subscript value cannot be negative

file1\_topic\_57: goes, computed, control, transferred, destination

- In both the cases the control is transferred subsequently to the statementx
- 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
- 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
- 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\_58: expression, returned, evaluated, conditional, nonzero

- 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
- Otherwise expression is evaluated and its value is returned
- It return zero otherwise

file1\_topic\_59: 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

file1\_topic\_60: step, endif, sum, palindrome, goto

- 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 then 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 Start Declare n r sum i ii False True Is i False r n mod of True Is False sumsumr tempsum nn Display Display Palindrome Palindrome Compiled by Sudip Lama Source code include while(n) include void main() rn sumsumr int nirsumtemp nn i clrscr() if(sumtemp) while(i) printf( Palindrome dtemp) else printf(nEnter the numbers) printf(Not Palindrome dtemp) sum tempn 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) printf(n The sum of series is fsum) Compiled By Sudip Lama void main() int n printf(nEnter the number of term) 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 SUM IF(IN) SUMSUMXP PP II GOTO PRINT SUM IS SUM Compiled By Sudip Lama Logical IF statement The logical if condition checks any given logical condition and transfer the control accordingly
- 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 WRITE ( ) FS TFS IF(TGT) GOTO WRITE ( ) T GOTO STOP IFTHENELSE statement The IFTHENELSE statement is more useful and easy to handle than the logical if statement
- That is IF (condn) THEN 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 IF(NEQ) THEN WRITE ( ) F WRITE ( ) FS DO IN TFS WRITE ( ) T CONTINUE ENDIF 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 Compiled By Sudip Lama IF(IREQ) THEN PRINT THE HCF ISI ENDIF II IIR GOTO Nested IFTHENELSE In certain cases we may have to use one IFTHENELSE structure within another IFTHEN
- IF (condn) THEN IF( condn) THEN IF(condn) THEN ENDIF ENDIF IF (condn) THEN ENDIF ENDIF ELSEIFTHEN structure The general format is IF (condn) THEN ELSE IF (condn) THEN Compiled By Sudip Lama ELSE IF(condn) THEN 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 PRINT TEMPIS NOT ARMSTRONG ENDIF 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 PRINT NUMIS NOT PRIME ENDIF Rules for subscripted variables The following rules may be strictly followed while defining the subscripted variables

## file1\_topic\_61: matrix, print, step, enter, array

- 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) then 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 Compiled by Sudip Lama Source code for(ii include for(jj(n c) False cc Display Display Compiled By Sudip Lama Array Array is a collection of variables of the same type that are referenced by a common base

- 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) Compiled By Sudip Lama for(iii) for(jijj) tempni 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

- 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) stdsqrt(stdsize) return std void main() float a i printf(nEnter element) for(iii) printf(n The standard deviation is fstar(a)) getch() Passing Two dimensional Array to the function The rules are Compiled By Sudip Lama

- Matrix Multiplication include include void input(int aint rint c) int ij for(iiri) for(jjcj) void display(int aint rint c) int ij for(iiri) for(jjcj) printf(n) void multiply(int aint bint cint rint cint c) int ijk for(iiri) for(jjcj) for(kkck) void main() int abcrcc 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) Compiled By Sudip Lama printf(nMultipication is not possible) 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

- 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 Compiled By Sudip Lama 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 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

- 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) Write a program to arrange one dimensional array in ascending order in FORTRAN

- C ASCENDING ORDER OF MATRIX Compiled By Sudip Lama INTEGER A()TEMP PRINT ENTER THE ELEMENT TO ARRAY DO I DO JI IF(A(I)GTA(J)) THEN TEMPA(I) A(J)TEMP ENDIF CONTINUE PRINT THE ELEMENT TO ARRAY WRITE () (A(I)I) Implied DO loop for multidimensional array The implied DO loop can also be used for multidimensional arrays

- 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 PRINT THE ELEMENTS OF ARRAY IS Write a program to add two matrixes and display the resultant matrix in FORTRAN

- Compiled By Sudip Lama 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 CONTINUE PRINT RESULTANT ARRAY IS DO IR DO JC PRINT C(IJ) CONTINUE PRINT ADDITION IS NOT POSSIBLE ENDIF Write a program perform matrix multiplication using nested DO loop C MATRIX MULTIPLICATION 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) Compiled By Sudip Lama CONTINUE PRINT ENTER ELEMENT TO SECOND ARRAY DO IR DO JC READ B(IJ) CONTINUE DO IR DO JC C(IJ) DO KR CONTINUE PRINT RESULTANT ARRAY IS DO IR DO JC PRINT C(IJ) CONTINUE PRINT MULTIPLICATION IS NOT POSSIBLE ENDIF Write a program to perform

matrix multiplication using implied DO loop C MATRIX MULTIPLICATION 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 PRINT ENTER ELEMENT TO SECOND ARRAY DO IR DO JC C(IJ) DO KC CONTINUE PRINT RESULTANT ARRAY IS PRINT MULTIPLICATION IS NOT POSSIBLE Compiled By Sudip Lama ENDIF Write a program to sum the following series up to n term PRINT ENTET THE VALUE OF N AND X READ NX SIGN FACT SUM DO IN SIGNSIGN DEN DO JFACT DENDENJ CONTINUE NUMSIGNXP SUMSUMNUMDEN FACTFACT PP CONTINUE PRINT THE SUM ISSUM Additional Topic Look for Preprocessor and Dynamic Memory Allocation

file1\_topic\_62: array, specifies, elements, hold, arraynamesize

- The array is given a name and its elements are referred by their subscripts or indices
- 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 general form of an array declaration is as shown below type arraynamesize Where type declares the data type of the array which is the type of each element in the array
- The arrayname specifies the name with which the array will be referenced and size defined how many elements the array will hold
- For example defines the arrays string as a five elements array
- All we have to do is to declare an array variable and use it
- array\_nameimember\_name Example classroll Practice Define a structure to hold the roll no

file1\_topic\_63: comprised, array, element, single, homogeneous

- 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
- 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\_64: columns, rows, varies, array, number

- 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 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

### file1\_topic\_65: initialization, initialized, dimensional, size, based

- 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 Compiled By Sudip Lama Initialization two dimensional array Two dimensional arrays are also initialize in the same ways as singledimension ones
- In such cases the size of array will be determined automatically based on the number of elements initialized

### file1\_topic\_66: 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\_67: structure, argument, actual, copy, member

- We cannot 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)` Compiled By Sudip Lama 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, struct, sec, min, thrthrthr

- Compiled By Sudip Lama 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 tsectsec include thrtmin struct time t mintmin return t int hr int min struct time sub(struct time t struct time t) int sec struct time t void input(struct time t) tsectsectsec t mintmintmin printf(nEnter the hr min and sec) thrthrthr scanf(dddthrtmintsec) if(tsec) void display(struct time t) tmin tsec printf(nhrdtmindtsecdtthrtmintsec) if(tmin) struct time add(struct time t struct time t) thr struct time t mintmin tsectsectsec t mintmintmin return t



thrthrthr tminsec 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 subtraction of time is) display(t) getch() Compiled By Sudip  
Lama Array of Structures We use structure to describe the format of a number of related  
variables

### file1\_topic\_69: opened, file, trying, operation, safe

- When we open a file we must specify what we want to do with the file
- While opening the file in this mode if file exists then the file is opened with the current content safe otherwise an error occurs ie fopen() returns NULL
- Trying to use a file that has not been opened
- Trying to perform an operation on a file when the file is opened for another type of operation

### file1\_topic\_70: rewind, file, points, opened, mode

- In this mode FILE pointer points to the starting of byte of file
- When the file is opened in this mode FILE pointer points to end of file
- Rewind takes a file pointer and resets the position to the start of the file
- rewind nd(fp) nft(fp) Would assign to n because the file position has been set to the start of the file be rewind
- Remember that whenever a file is opened for reading or writing a rewind is done implicitly

### file1\_topic\_71: exist, created, file, deleted, invalid

- In this mode a file with specified name is created if the file does not exist
- The contents are deleted if the file already exists
- A file with the specified name is created if the file does not exist
- Opening a file with an invalid filename

### file1\_topic\_72: 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\_73: feof, ferror, file, end, message

- We have two status inquiry 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
- 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
- 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
- The statement if(ferror(fp)) printf(An error has occurred) Would print the error message if the reading is no successful
- This function helps us in reading a file more than once without having to close and open the file

### file1\_topic\_74: position, current, offset, fseek, takes

- 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
- fseek function is used to move the file position to a desired location within the file
- It takes the following form fseek(file\_ptr, offset, position) 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(fp, L) meaning go to begining refer to class note for example Compiled By Sudip Lama FORTAN character set The following is the set of character used in FORTRAN

file1\_topic\_75: exponent, mantissa, exponential, form, alphabet

- Compiled By Sudip Lama Exponent Form The following rules apply to the exponential form of real constant
- The exponential form has two parts (i) mantissa (ii) exponent
- The alphabet E is written in between the mantissa and the exponent
- The exponent is always an integer with at most two digit

file2\_topic\_76: 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\_77: step, total, mark, sheet, division

- 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\_78: assisting, examine, dynamics, guides, himher

- 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

file2\_topic\_79: true, operands, false, logic, result

- 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
- It is used when there are two possible actions one when a condition is true and the other when it is false

file2\_topic\_80: inputoutput, input, keyboard, conio, device

- InputOutput The inputoutput symbol is used to denote any function of an inputoutput device in the program
- Reading input data from keyboard and displaying the output data on screen such input output system is considered as conio input out
- 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

file2\_topic\_81: executed, consecutive, processing, order, arithmetic

- Thus all arithmetic processes of adding subtracting multiplying and dividing are shown by a processing symbol
- 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 other word consecutive addition and subtraction operations are carried out from left to right as are consecutive multiplication division and remainder operations
- 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
- ) Increasing the execution speed as they refer address

file2\_topic\_82: bytes, byte, address, allocated, memory

- The logical process of moving data from one location of the main memory to another is also

denoted by this symbol

- 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
- Downloaded from www.jayaram.comnp Downloaded from www.jayaram.comnp 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)
- The integer will occupy bytes
- The integer will take only bytes in because in a text file it is stored in a binary file because in binary file it is sequence of characters ie

## file2\_topic\_83: 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\_84: documentation, internal, explanatory, comments, meaningful

- Program Documentation Program Documentation refers to the details that describe a program While writing programs it is good programming practice to make a brief explanatory note on the program or program segment This explanatory note is called comment It explains how the program works and interact with it Thus it helps other programmers to understand the program There are two types of documentation Internal documentation External documentation

- 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 program It may be message to the user to respond to the program requirement This is accomplished using output statements It makes a program more attractive and interactive Some 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
- It can also be defined as a section of a program performing a specific task

file2\_topic\_85: language, unix, level, high, laboratories

- 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
- It also resembles other high level structure programming language such as Pascal and Fortran
- Historical Development of C C was an offspring of the Basic Combined Programming Language (BCPL) called B and was implemented at Bell Laboratories in
- 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 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
- C was now standardized by American National Standard Institute
- Such type of C was named ANSI C Importance of C Nowadays the popularity of C is increasing probably due to its many desirable qualities
- 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
- In fact many of the C compilers available in the market are written in C Programs written in C are efficient and fast
- It is many times faster than BASIC (Beginners All Purpose Symbolic Instruction Code a high level programming language)
- C is highly portable
- This means that C programs written for one computer can be seen on another with little or no modification
- C Language is well suited for structure programming thus requiring the user to think of a problem in terms of function modules or blocks
- Another important feature of C is its ability to extend itself

file2\_topic\_86: access, takes, time, programs, random

- for writing operating systems as well as for applications programming such as for writing a program to solve mathematical equation or for writing a program to bill customers)
- Several standard functions are available which can be used for developing programs
- accessed at any given time
- All these details are hidden from the programmer and the operating system takes care of all these things
- So far we have used only sequentially access in our programs
- In random access data can be accessed and processed randomly ie
- Random access takes less time than the sequential access
- specific programs written for them
- It take more time for ip output operation

- It takes less time for ip output operation compared to binary file

## file2\_topic\_87: section, collection, functions, modules, divided

- 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
- A C program may contain one or more sections shown in Documentation Section Link Section Global Declaration Section Main() Function Section Declaration Part Execution Part Subprogram Section Function Function (Userdefined Functions) Function n Fig Basic Structure of a C program Downloaded from wwwjayaramcomnp Downloaded from wwwjayaramcomnp The documentation section consists of a set of comment lines giving the name of the section provides instructions to the compiler to link function from the system library
- 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
- A complex C program can be divided into a number of user defined functions

## file2\_topic\_88: prototypes, ahead, function, main, userdefined

- 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
- 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
- Arun Kumar Yadav Lecturer Eastern College of Engineering Biratnagar Computer Programming in C main( ) int c double d clrscr( ) 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
- 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
- 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
- include include void main( ) Er

file2\_topic\_89: symbolic, constants, constant, numeric, sequence

- The definition defines all the symbolic constants
- They are often referred to collectively as numeric \_ type constants
- Commas and blank spaces cannot be included within the 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

file2\_topic\_90: changed, actual, passed, persists, mechanism

- There are some variables that are used in more than one function
- But actual parameters are not changed

- its value persists between different function calls
- 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

#### file2\_topic\_91: global, variables, local, destroyed, exited

- 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
- 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
- What are local and global variables

#### file2\_topic\_92: 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\_93: 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\_94: img, real, fabs, area, sqrt

- Arun Kumar Yadav Lecturer Eastern College of Engineering Biratnagar Computer Programming in C First Program written in C Save it as helloc include header file main ( ) main ( ) function Print (Hello World n) statement output Hello World Program to calculate the area of a circle areac include library file access main( ) function heading float radius area variable

declaration clrscr() Console function Used to clear screen printf (Enter radius) output statement scanf (f radius) input statement area radiusradius assignment statement printf (Area of area) output statement getch() Console function Used to hold the same screen Some Important Questions C language is a middle level language

- x area \_\_ temperature PI

- Downloaded from www.jayaram.comnp Downloaded from www.jayaram.comnp 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 input/output statement of C programming

- Arun Kumar Yadav Lecturer Eastern College of Engineering Biratnagar Computer Programming in C printf(d f c nxyz) getch() Write short notes on Formatted input/output function

- Write the input/output formatted function along with syntaxes Downloaded from www.jayaram.comnp Downloaded from www.jayaram.comnp Control Statements The statements which alter the flow of execution of the program are known as control statements

- Arun Kumar Yadav Lecturer Eastern College of Engineering Biratnagar Computer Programming in C printf(Enter value of x) for(I term used for math function like sqrt() and fabs() main() Float a b c xxdreal img clrscr() Er

- Arun Kumar Yadav Lecturer Eastern College of Engineering Biratnagar Computer Programming in C 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 ) 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

- Arun Kumar Yadav Lecturer Eastern College of Engineering Biratnagar Computer Programming in C getch() The output of above program is Because the statement printf(dn(pi)) execute times from i to i

file2\_topic\_95: set, letters, building, eliminates, numerically

- 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

- 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)

- What do you mean by Character set in C

file2\_topic\_96: group, whitespace, sign, percent, characters

- The special characters are listed below (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

- Each character group must begin with a percent sign (%)

- In its simplest form a single character group will consist of the percentage sign followed by a conversion character which indicates the types of corresponding 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

- An individual character group in control string will consist of the percent sign followed by a conversion character indicating the type of the corresponding data item

- Multiple character group can be contiguous or they can be separated by other characters including whitespace character

- The use of blank spaces as character group separators is particularly common

- They must be placed immediately after the percent sign

file2\_topic\_97: tokens, punctuation, passage, token, individual

- 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

- What are the token of C language

## file2\_topic\_98: lowercase, letters, underscore, upper, 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)
- 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
- First character must be an alphabet (or Underscore)
- 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
- To do so the sign within the appropriate control group is followed by an asterisk ()
- It must start with a period ()
- Write a program to convert a lowercase character string into uppercase using array

## file2\_topic\_99: bytes, byte, signed, unsigned, needs

- 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 Type | 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

- Arun Kumar Yadav Lecturer Eastern College of Engineering Biratnagar Computer Programming in C A memory space equivalent to times the size of an integer (ie
- Because I needs bytes f needs bytes and c needs byte bytes is highest in these

## file2\_topic\_100: types, basic, datatype, primary, fundamental

- There are three cases of data types
- Basic data types (Primary or Fundamental) eg
- Derived data types eg
- User defined data types eg
- The basic data types are listed below
- What is a datatype
- Define the data types
- Describe different data types

## file2\_topic\_101: base, constants, floating, point, decimal

- Integer and floating point constants represent numbers
- The following rules apply to all numeric type constants
- 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
- Arun Kumar Yadav Lecturer Eastern College of Engineering Biratnagar Computer Programming in C XFFFFFFUL 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

## file2\_topic\_102: logical, expressions, referred, interpretation, relational

- The minus sign is an operator that changes the sign of a positive constant though it can be  
thought of as a part of the constant itself
- Several relational expressions involving these variables Expressions Interpretation Value i k  
true (jk) (i) false k

- This operator is referred to as the logical negative (or logical not) operator
  - not Logical operators are used to compare evaluate logical and relational expressions
  - Operator is referred as logic and the operator is referred as logic or
  - 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

## file2\_topic\_103: constants, octal, integer, incorrectly, digits

- Thus it consists of a sequence of digits
- A decimal integer constant can consists of any combination of digits taken from the set through
- If the constant contains two or more digits the first digit must be something other than
- Several valid decimal integer constants are shown below The following decimal integer constants are written incorrectly for reason stated Illegal character ( ) Illegal character ( )
- 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
- Valid octal number (integer) constants are shown below Downloaded from wwwjayaramcomnp Downloaded from wwwjayaramcomnp The following octal integer constants are written incorrectly for the reason stated Does not begin with
- A hexadecimal integer constant must begin with either or X
- 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 1 though they may not be negative
- The general form represents an octal digit ( through )

## file2\_topic\_104: illegal, character, exponent, blank, xbff

- Illegal character (blank space ) Illegal character ( ) the first digit cannot be zero
- Illegal character ( ) Illegal character ( )
- xbff Illegal character ( )
- Illegal character ( ) E The exponent must be an integer (it cannot 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\_105: unsigned, appending, long, identified, constant

- An unsigned integer constant can be identified by appending the letter ( u ) (either upper or lowercase) to the end of the constant
- 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 precede 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) X U hexadecimal (unsigned) Er

file2\_topic\_106: escape, sequence, sequences, backslash, newline

- Thus the string would be displayed Line Line Line The compiler automatically places a null character ( \0 ) at the end of every string constant as the last character within the string (before the closing double quotation mark)
- c a l i f o r n i a Subscript Escape Sequence Certain nonprinting character as well as the backslash ( \ ) and apostrophe ( ' ) can be expressed in terms of escape sequences
- 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 ( ? )  
- backslash ( \ ) null Several character constants are expressed in terms of escape sequences are The last three escape sequences represent an apostrophe backslash and a quotation mark respectively
- Escape Sequence represents the null character (ASCII \0) 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 ( 0 through 9 and a through f)
- What is Escape Sequence



- Write any four escape sequence with meaning and symbols
- List out any three escape 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
- Normally zero is used to indicate normal termination and non zero value to indicate termination due to some error or abnormal condition

file2\_topic\_107: type, items, data, class, belong

- However the data type associated with the variable are not change
  - The data items must all be of the same type (eg
  - The existing data type may belong to any class of type including the user defined
- Downloaded from www.jayaram.comnp Downloaded from www.jayaram.comnp ones
- The new type is new only in name but not the data type
  - 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
  - data\_type is basic data type
  - The data item can be different type some can be int some can be float some can be char and so on

file2\_topic\_108: dddnabc, following, abc, double, line

- A C program contains the following lines int abc Er
  - 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 d n sizeof(int)) and so on
  - Arun Kumar Yadav Lecturer Eastern College of Engineering Biratnagar Computer Programming in C double x y printf (ffffn 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 (cid) Flags Optional The flag affect the appearance of the output

- (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(df e nxyz) Er

file2\_topic\_109: root, declarations, contains, short, long

- Arun Kumar Yadav Lecturer Eastern College of Engineering Biratnagar Computer Programming in C 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
- Downloaded from wwwjayaramcomnp Downloaded from wwwjayaramcomnp 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
- A C program contains the following type declarations
- int a is a declaration of one dimensional array of type int

file2\_topic\_110: assigned, earlier, replacing, quantity, lines

- 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
- ) is assigned to c replacing the earlier value
- Finally the character w is assigned to d replacing the earlier character a

## file2\_topic\_111: 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` cannot 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\_112: cprogram, types, various, passed

- Describe different data types are used in cprogram
- Write down the various types of it in Cprogram
- `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 they declared in Cprogram

## file2\_topic\_113: operand, operands, operators, requires, operator

- Some operators require two operands while other act upon only one operand
- 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 cannot be used in place of one another
- The operand used with each of these operators must be a single variable
- Bitwise operators can be applied only to integer type (signed or unsigned) and not to float or double

## file2\_topic\_114: 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\_115: bitwise, right, left, shift, bits

- The operators are grouped hierarchically according to their precedence
- 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
- Arun Kumar Yadav Lecturer Eastern College of Engineering Biratnagar Computer Programming in C 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
- Inequality Bitwise AND Left Right Bitwise XOR Left Right Bitwise XOR Left Right Logic AND Left Right Logic OR Left Right

## file2\_topic\_116: operand, altered, utilized, compile, qualifier

- 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

## file2\_topic\_117: follow, assign, assigned, value, example

- a b In this example x will be assigned the value of b
- The value assign like follow a a a a a Er
- Arun Kumar Yadav Lecturer Eastern College of Engineering Biratnagar Computer Programming in C 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 int a int a would never work

## file2\_topic\_118: num, number, prime, odd, end

- Write a program to enter two numbers

- Arun Kumar Yadav Lecturer Eastern College of Engineering Biratnagar Computer Programming in C while condition Body of Loop Next statement out of loop fig Flowchart of while loop Program to print the sum of digits of any num 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( ) printf (dt i) while (i) printf(n) getch( ) Output Er

- 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 Downloaded from wwwjayaramcomnp Downloaded from wwwjayaramcomnp 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) Er

- Arun Kumar Yadav Lecturer Eastern College of Engineering Biratnagar Computer Programming in C scanf (d num) remainder num modular division if (remainder ) test for even printf (Number is evenn) printf ( Number is oddn) getch( ) Output Enter a number Number is odd

- The general syntax continue Downloaded from wwwjayaramcomnp Downloaded from wwwjayaramcomnp Loop statements before continue iteration of continue statements condition True false Loop statements after 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 Er

- 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 goto odd even printf (Number is even) goto end Downloaded from wwwjayaramcomnp Downloaded from wwwjayaramcomnp odd printf (Number is odd) end printf (n) getch( ) Output Enter the number Number is even

- Arun Kumar Yadav Lecturer Eastern College of Engineering Biratnagar Computer

Programming in C printf (Enter your choice ) scanf (d choice) switch (choice) case printf (dataset 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 choice) 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

- program to check the num is prime or not primec include main( ) int numic clrscr ( ) printf(Enter number) Downloaded from wwwjayaramcomnp Downloaded from wwwjayaramcomnp scanf(dnum) ic while(inum) if(num i ) if(c ) printf(d is prime numbern num) printf(d is not prime numbernnnum) getch( ) Output Enter number is prime number write a program to calculate to n terms main( ) int nsumitem clrscr( ) printf(Enter how many tems ) for(I term i n i term ) sum term printf(n The is t dsum) getch() write a program to calculate xxxto n terms main( ) int nxsumitem clrscr( ) printf(Enter how many tems ) Er

- 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

- Arun Kumar Yadav Lecturer Eastern College of Engineering Biratnagar Computer Programming in C 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 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

- 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\_119: input, data, given, device, devices

- 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

- When a program needs data it takes the data through the input functions and sends the results to output devices through the output functions

## file2\_topic\_120: input, inputoutput, scanf, console, functions

- Input Program Output Fig Input Output To perform inputoutput operation in console mode C has a number of input and output functions
- The standard input functions are scanf( ) getch( ) getchar( ) gets( ) etc
- The standard library stdio.h provides functions for input and output
- Types of IO The inputoutput functions are classified into two types i Formatted functions ii
- 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
- 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)
- For example printf() scanf() sqrt() getch() etc
- Arun Kumar Yadav Lecturer Eastern College of Engineering Biratnagar Computer Programming in C Data Files The input output function Like printf( ) scanf( ) getchar( ) putchar( ) gets( ) puts( ) are known as console oriented IO functions which always use keyboard for input device
- Describe any two file handling inputoutput function

## file2\_topic\_121: scanf, arg, entered, maximum

- 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 scanf (d f c i x c) Downloaded from www.jayaram.comnp Downloaded from www.jayaram.comnp 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
- In contrast to scanf() function the arguments in a printf() function do not represent memory addresses and therefore are not preceded by ampersands
- It offers an alternative function of scanf( ) function for reading strings
- Unlike scanf( ) function it doesnot skip whitespaces

file2\_topic\_122: command, argv, argument, argc, vector

- 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
- 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
- include main(argc argv) main with arguments int argc argument count char argv list of arguments file fp int i char word fpopen(argv w) open file with name arg printf(n No arguments in command line dnn argc) for (i ilarge i) Er

file2\_topic\_123: alphabet, case, pal, palindrome, len

- Different methods for entering the inputs Biratnagar Biratnagar Biratnagar Biratnagar  
Downloaded from wwwjayaramcomnp Downloaded from wwwjayaramcomnp Example  
include main( ) char line scanf ( ABCDEFGHIJKLMNOPQRSTUVWXYZ line) If the string  
EASTERN COLLEGE OF ENGINEERING Is entered from the standard input device when the  
program is executed the entire string will be assigned to the array line since the string is



comprised entirely of uppercase letters blank spaces

- Arun Kumar Yadav Lecturer Eastern College of Engineering Biratnagar Computer Programming in C  
printf (Enter your valid operation) end of switch getch( ) end of main( )  
Program to find whether the alphabet is a vowel or consonant include main( ) char ch clrscr( )  
printf (Enter an alphabet ) scanf (c ch) switch (ch) case a case A case e case E case i case I case o  
case O case u case U printf (Alphabet is a vowel\n) break default printf (Alphabet is a constant\n)  
end of switch getch( ) end of main( ) Continue statement The continue statement is used to  
bypass the remainder of the current pass through a loop
- A program to read a string and check for palindrome void main( ) Downloaded from  
wwwjayaramcomnp Downloaded from wwwjayaramcomnp 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) printf (n the input string is  
palindrome) getch( ) Some Important Questions What is an array

## file2\_topic\_124: width, field, specified, item, 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 percent sign ( ) and the conversion character
- Arun Kumar Yadav Lecturer Eastern College of Engineering Biratnagar Computer  
Programming in C The data item may contain fewer characters than the specified filed width
- However the number of characters in the actual data item cannot 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 blank spaces required to fill the Downloaded from wwwjayaramcomnp Downloaded from  
wwwjayaramcomnp minimum field width will be added after the data item rather than the  
data item
- TableFlags (cid) 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\_125: flag, blank, flags, justified, precede

- The flags may be blank space or
- Flags Meanings Data item is left justified within the field
- A sign (either or ) will precede each signed numerical data item
- Without this flag only negative data items are preceded by a sign
- Applies only to data items that are right justified within a field whose minimum size is larger than the data item (blank space) A blank space will precede each positive signed numeric data item This flag is overwritten by the flag if both are present
- When flag is used with o or x it causes octal and hex items to be preceded by 0 and x respectively

file2\_topic\_126: file, source, content, argc, argv, welcome

- `printf (e ) e printf (g )` Downloaded from [www.jayaram.com](http://www.jayaram.com) Downloaded from [www.jayaram.com](http://www.jayaram.com) eg
- For `Downloaded from www.jayaram.com` Downloaded from `www.jayaram.com` instance for the command line given above `argc` is three and `argv` is an array of three pointers to strings as shown below `argv[0]` PROGRAM `argv[1]` X\_FILE `argv[2]` Y\_FILE In order to access the command line arguments we must declare the main function and its parameters as follows `main(argc argv)` `int argc char argv` WAP that will receive a filename and a line of text as command line arguments and write the text to the file
- 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("test.txt" w) if(fp NULL) 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 test.txt in notepad where the content is Welcome to Eastern College of Engineering`
- Arun Kumar Yadav Lecturer Eastern College of Engineering Biratnagar Computer Programming in C Program to open the file test.txt created above read its content display to the screen include `Void main( ) FILE fp char s fp fopen("test.txt" r) if (fp NULL) printf("In file can not be opened) exit( ) fgets(s fp) printf("in the text from file is %s" 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) printf(n Destination file cant be created) Downloaded from wwwjayaramcomnp Downloaded from wwwjayaramcomnp exit( ) while(( ch fgetc(source)
- 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) rewind(fptr) Er
- Arun Kumar Yadav Lecturer Eastern College of Engineering Biratnagar Computer Programming in C printf(The content from file n) while( cc fgetc fptr))
- 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\_127: repeated, execute, times, advance, repetitions

- Sometimes we may want to execute some statements several times
- For Loop For loops is useful to execute a statement for a number of times
- When the number of repetitions is known in advance the use of this loop will be more efficient
- Loop is used to do repeated task
- the same function to do repeated

file2\_topic\_128: 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\_129: loop, dowhile, loops, known, inside

- 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

- DOWHILE Each loop consists of two segments one is known as the control statement and the other is the body of the loop
- Depending on the position of control statement in the loop loops may be classified either entry\_controlled loop or exit\_controlled loop
- While and For are entry\_controlled loops whereas dowhile is exit\_controlled loop
- Thus this loop is also known as determinate or definite loop
- Each execution of the loop body is known as iteration
- For example a for loop may be nested inside another for loop or inside a while or dowhile loop

## file2\_topic\_130: loop, condition, body, dowhile, statementn

- There are three types of loop statements in C
- The general syntax for (counter initialization test condition increment or decrement) body of loop Flowchart of For Loop Er
- The general syntax is Downloaded from wwwjayaramcomnp Downloaded from wwwjayaramcomnp do do statement while(condition) statement statement statementn while(condition) Flowchart of dowhile is Body of Loop False Condition Next statement out of loop fig flowchart of dowhile loop Here firstly the segments inside the loop body are executed and then the condition is evaluated
- Describe the different types of looping statement used in C with flow chart

## file2\_topic\_131: factorial, fact, num, number, calculated

- Arun Kumar Yadav Lecturer Eastern College of Engineering Biratnagar Computer Programming in C Counter Initialization Test False Condition Body of Loop Update expression Out of Loop Fig Flowchart of For Loop Algorithm Program to calculate factorial Start
- Print Enter a number whose factorial is to be calculated
- Initialize fact to 1 and counter i to 1 For inum factfacti End of For Print fact as factorial of the number num
- Stop Downloaded from wwwjayaramcomnp Downloaded from wwwjayaramcomnp 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

- Arun Kumar Yadav Lecturer Eastern College of Engineering Biratnagar Computer Programming in C 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 ) scanf (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
- WAP to find the factorial of a number using recursive method long int factorial (int n) if (n ) return ( ) 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 Recursion Iteration
- Write a program to calculate factorial of n number by using recursive function where n is the number inputted by user
- Determine what would be the output of the following int il printf(id id idi i i) What is the method of declaring local and global variables
- Write a program in C to allow a user to enter an integer number interactively and display its factorial value

file2\_topic\_132: body, condition, loop, executed, true

- 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
- If the condition is true then again the loop body is executed and this process continues until the condition becomes false
- condition is evaluated first and body of loop the body of the loop is executed first without is executed only if this test is true
- checking condition and at the end of body of loop the condition is evaluated
- The body of the loop is always executed at at all if the condition is not satisfied at the least once
- Instead the remaining loop statements are skipped and the computation proceeds directly to the next pass through the loop

file2\_topic\_133: loop, terminate, completely, exit, come

- test dowhile loop is exit controlled loop ie
- The body of the loop may not be executed
- 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

## file2\_topic\_134: condition, year, statement, large, leap

- 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) statement statement n There can be a single statement or a block of statements after the if part
- Arun Kumar Yadav Lecturer Eastern College of Engineering Biratnagar Computer Programming in C Condition Statement False Next Statement 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) statement else statement statement statement Flowchart Condition statement statement Next statement 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
- For example if(condition) statementA statement A Here we have ifelse inside both if block and else block if(condition) statementB statementB Program to find whether a year is leap or not include main( ) int year clrscr( ) printf (Enter year ) scanf (d year) Downloaded from wwwjayaramcomnp Downloaded from wwwjayaramcomnp if(year ) if(year ) printf (Leap year n) 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) 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 large c if (bc) large b large c printf (Largest number is dn large) getch( ) End of main( ) Er
- if(condition) statementA statementA elseif(condition) statementB statementB elseif(condition) else statementC if (condition) else statementC statement D statement D The flowchart for else if statement is False Condition False Condition

Stat A False Condition Stat B Stat D Stat C Next Statement Fig Flowchart of else if ladder  
Downloaded from www.jayaram.comnp Downloaded from www.jayaram.comnp Program to find  
out the grade of a student when the marks of subjects are given

- Arun Kumar Yadav Lecturer Eastern College of Engineering Biratnagar Computer Programming in C 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
- some condition will be satisfied

file2\_topic\_135: matches, case, block, nested, executed

- 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 if else statement
- This type of nesting is frequently used in programs and is also known as else if ladder
- If the value of expression matches with any case constant then all statements under that particular case are executed
- If none of the case constant matches with the value of the expression then the block of statements under default is executed
- The statements of case and default are also executed in addition to the statements of case

file2\_topic\_136: recordn, delete, insert, recordsn, databasen

- Create databasen) printf (
- Insert new recordn) printf (
- Modify a recordn) printf (
- Delete a recordn) printf (
- Display all recordsn) printf (
- r) printf(Name t Marks n ) Er

file2\_topic\_137: parameter, type, mul, return, parametern

- The return type argument and body of the function are defined by the programmer as required
- A function can return value of any data type
- The general syntax is return\_type function\_name (parameter parameter parametern) statements where (cid) return\_type is the data type specifier of data returned by the function

- The syntax of the 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

file2\_topic\_138: cid, body, parenthesis, subscript, formal

- (cid) function\_name is the identifier by which it will be possible to call the function
- (cid) statements is the functions body
- (cid) 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
- When we call the function no need to subscript or square brackets

file2\_topic\_139: return, age, funct, height, terminate

- Arun Kumar Yadav Lecturer Eastern College of Engineering Biratnagar Computer Programming in C return statement The return statement is used in a function to return a value to the calling function
- There are two ways in which it can be used return return (expression) where return is a keyword
- The first form of return statement is used to terminate the function without returning any value
- In this case only return keyword is written
- 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
- The value returned by the return statement may be any constant variable expression or even any other function call which returns a value

file2\_topic\_140: sum, arguments, return, add, functions

- 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 arguments and a return value Er

- Arun Kumar Yadav Lecturer Eastern College of Engineering Biratnagar Computer Programming in C 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 Downloaded from wwwjayaramcomnp Downloaded from wwwjayaramcomnp int func(void) return (expressions) Program to illustrate function with no argument and a return value include 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

- Arun Kumar Yadav Lecturer Eastern College of Engineering Biratnagar Computer Programming in C 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

file2\_topic\_141: address, swap, passing, local, void

- Arun Kumar Yadav Lecturer Eastern College of Engineering Biratnagar Computer Programming in C printf (ydn y) getch( ) void swap (int a int b) function definition int t t a a b b t Output x y Function call by reference In this type of function call the address of variable or argument is passed to the function as argument instead of actual value of variable

- 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 y Downloaded from wwwjayaramcomnp

Downloaded from [www.jayaramcomnp.com](http://www.jayaramcomnp.com) clrscr( ) swap (x y) function call by address printf (x &n x) printf (y &n y) getch( ) void swap(int a int b) int t t a a b b t Output x 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

- Give the syntax of declaring a function in C Explain how an array can be passed to a function with the help of a suitable program in C What is Differences between call by value and call by reference

- Arun Kumar Yadav Lecturer Eastern College of Engineering Biratnagar Computer Programming in C 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

- Program to illustrate the use of passing pointer to a function include void addGraceMarks(int m) m m void main( ) int marks printf(Enter actual marks It) scanf(&marks) add GraceMarks(&marks) Passing address printf(n The graded marks is %d marks) Program to convert upper case letter into lower and vice versa using passing pointer to a function Downloaded from [www.jayaramcomnp.com](http://www.jayaramcomnp.com) Downloaded from [www.jayaramcomnp.com](http://www.jayaramcomnp.com) include void conversion(char) function prototype main( ) char input clrscr( ) printf(Enter character of our choicen) scanf(&input) conversion(&input) printf(n The corresponding character is %c\n input) getch( ) void conversion(char\* c) if(c c c) Output Enter character of our choice a The corresponding character is A Relationship between Arrays and Pointers There is a close association between pointer and array

file2\_topic\_142: 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) Downloaded from [www.jayaramcomnp.com](http://www.jayaramcomnp.com) Downloaded from [www.jayaramcomnp.com](http://www.jayaramcomnp.com) void main( ) clrscr( ) printf (tda)

func( ) printf (tda) Output Illustration 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\_143: memory, performance, considerably, slower, associates

- Storage class refers to the performance of a variable and its scope within the program
- Automatic variables are allocated storage in the memory of the computer however for most computers accessing data in memory is considerably slower than processing in the CPU
- 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\_144: static, initialized, variable, programin, retained

- 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 0
- Program with auto variable Program with static variable void increment( ) void increment( )  
int i static int i printf (dn i ) printf (dn i ) Er
- What is a static variable
- Write a program in C that uses a static variable

file2\_topic\_145: calloc, malloc, memory, allocated, realloc

- These storage cells are called registers
- They are allocated storage upon entry to a block and the storage is freed when the block is exited
- include Downloaded from www.jayaram.comnp Downloaded from www.jayaram.comnp  
include void main( ) int a int i j clrsc( ) for(c i i) for(jo j j) printf(n) getch( ) output Dynamic  
Memory Allocation (DMA) The process of allocating and freeing memory at run time is ka  
Dynamic Memory Allocation
- 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
- In such situation DMA will be useful
- This is ka DMA
- DMA refers allocating and freeing memory at run time
- There are library functions malloc( ) calloc( ) free( ) and realloc( ) for memory management
- These functions are defined within header file stdlib.h 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\_type 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
- For example x(int)malloc(sizeof(int)) Er
- ) 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
- Thus it is normally used for requesting memory space at runtime for storing derived data type such as arrays user defined
- Its syntax 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
- ) free( ) The builtin function frees previously allocated space by calloc malloc or realloc function
- The memory dynamically allocated is not returned to the system until the programmer returns the memory explicitly
- This can be done using free( ) function
- This function is used to release the space when it is not required
- 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

- ) realloc( ) This function is used to modify the size of previously allocated space
- Sometimes the previously allocated memory is not sufficient we need 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( )
- Its syntax is as If the original allocation is done by the statement ptrmalloc(size) then reallocation of space may be done by the statement ptrrealloc(ptrnewsize) Downloaded from www.jayaramcomnp Downloaded from www.jayaramcomnp 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
- Program to illustrate the use of realloc( ) include include include void main( ) char name[100]; clrscr( ) name[0]='\0'; malloc(100); strcpy(name,"Jayaram");

## file2\_topic\_146: pole, disks, disk, transfer, temp

- 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
- Each disk has a hole in the centre allowing it be stacked
- 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
- left Center Right 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 rightmost pole using the centre pole for immediate storage
- The Tower of Hanoi\_solved using recursion include void transfer (int n char from char to char temp) function prototype main( ) Downloaded from wwwjayaramcomnp Downloaded from wwwjayaramcomnp 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 return(fib(x) fib(x)) main() int i n no clrscr( ) printf (how many no in series ) scanf(d n) Er

## file2\_topic\_147: sort, numbers, ascending, continuous, locations

- Suppose we have numbers of type integer and we have to sort them in ascending or descending order
- 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 ) for (i i i) Er
- Arun Kumar Yadav Lecturer Eastern College of Engineering Biratnagar Computer Programming in C 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
- Write a program to accept numbers and sort them with use of pointer
- Write a program to read numbers and sort them in ascending order using pointer

## file2\_topic\_148: inefficient, elements, different, illustrated, increase

- If we have no array we have to define different variable like a a a of type int to store these

twenty numbers which will be possible but inefficient. If the number of integers increase the number of variables will also be increased and defining different variables for different numbers will be impossible and inefficient.

- Its elements can be illustrated: 1st element, 2nd, 3rd, 4th, 5th element, a, a, a, a, a. The elements of an integer array are stored in continuous memory locations.
- What are the advantages of using array?
- Through arrays can be used for data storage they are of fixed size.
- we can store arrays each of elements of integer types.

## file2\_topic\_149: matrix, enter, second, int, column

- index. For example, in `int a` we can write the following statement: `a[x]`. `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 = 0; i < n; i++)`.  
Downloaded from [www.jayaram.com](#)  
Downloaded from [www.jayaram.com](#)  
`clrscr( )` for (i = 0; i < n; i++) {  
main( ) int matrix[10][10];  
scanf(d ai j) printf(Enter first matrix n) for (i = 0; i < n; i++) for (j = 0; j < n; j++) scanf(d bi j) Er

- Arun Kumar Yadav Lecturer Eastern College of Engineering Biratnagar Computer Programming in C  
for (i = 0; i < n; i++) {  
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) scanf(d l)  
Downloaded from [www.jayaram.com](#)  
Downloaded from [www.jayaram.com](#)  
printf(Enter column of second matrix() t) scanf(d p) if (nl) printf(Multiplication is not possible ) printf(Enter the first matrix n) for (i = 0; i < m; i++) for (j = 0; j < n; j++) printf(Enter a d t i j) scanf (d a i j) printf(Enter the second matrix n) for (i = 0; i < l; i++) for (j = 0; j < p; j++) printf(Enter b d t i j) scanf(d bi j) for (i = 0; i < m; i++) for (j = 0; j < p; j++) {  
si = 0; for (k = 0; k < n; k++) si += a[i][k] \* b[k][j];  
printf (The matrix multiplication is n) for (i = 0; i < m; i++) for (j = 0; j < p; j++) printf(dt si j) printf(n) end of else  
Er

- Arun Kumar Yadav Lecturer Eastern College of Engineering Biratnagar Computer Programming in C  
`getch( )` end of `main( )`  
Output: Enter row of the first matrix ( ) Enter column of the first matrix ( ) 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 a b ( ) s a b e f A B c d g h aebg af bh AB cedg cg dh)  
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.  
Downloaded from [www.jayaram.com](#)  
Downloaded from [www.jayaram.com](#)  
`main( )` int num; `clrscr( )` printf (n The content of array is n) for (i = 0; i < n; i++) {  
include `int sum (int a )` void output (int a )

main( ) int a s i clrscr( ) printf (Enter elements n) for (i i i) scanf (d ai) output (a) s sum (a) printf (Sum of array element is n s) getch( ) int sum (int a ) for (i i i) n nai return (n) Er

- Arun Kumar Yadav Lecturer Eastern College of Engineering Biratnagar Computer Programming in C void output (int a ) int i printf (The elements of array are 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 ) input(a) Downloaded from wwwjayaramcomnp Downloaded from wwwjayaramcomnp 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 Er

- Write a program to sum the diagonal element of a square matrix of order n Write a program that read a matrix and find sum of all the elements of matrix then print the sum

- include include main( ) int a int P i clrscr( ) p a for(i i i) Er

- 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) getch( ) output Pointer and Multidimensional Array We know that the name of array is point to first element of array ie

- Write a program to calculate sum of all number in a given matrix using pointer

- Write short notes on Dynamic memory allocation Write a function using pointers to add two matrices

- Arun Kumar Yadav Lecturer Eastern College of Engineering Biratnagar Computer Programming in C Roll Marks If two members are used simultaneously the output is unexpected as following

file2\_topic\_150: length, null, maximum, excluding, terminator

- In other words we can say that a string is a sequence of contiguous character in memory terminated by the null character

- 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\_151: files, level, high, binary, buffer

- These problems invite the concept of data file in which data can be stored on the disks and read whenever necessary without destroying data



- A file is a place on the disk where a group of related data is stored
- 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
- 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 explicitly in case of lower level 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
- For each binary and text files there are a number of formatted and unformatted library functions in C Data files High Low Text Binary fig classification of files Downloaded from [www.jayaramcomnp](#) Downloaded from [www.jayaramcomnp](#) 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 stdio.h that contains all information about the file like name status buffer size current position and of file status etc
- Data is stored on the disk in the same way as it is represented in computer memory
- Text files are in human readable form and
- Binary files are not in human readable they can be created and read using any form and can be created and read by text editor
- (PU) What is a files ad explain its importance in C programming
- What is a file

## file2\_topic\_152: graphics, mode, text, dots, shapes

- Arun Kumar Yadav Lecturer Eastern College of Engineering Biratnagar Computer Programming in C Graphics The C graphics function fall into two categories those that work in text mode and those that work in graphics mode
- The text mode functions are concerned with placing text in certain area of screen
- Graphics mode function allow us to draw dots lines and shapes (like circle rectangle etc
- In order to perform graphics related activities we include graphics.h

- Some text mode graphics function `clrscr()` This function erases the text window
- The syntax of `closegraph()` function 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 complex shapes
- In text mode we can address only location `()` but in graphics mode 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\_153: graphics, mode, initgraph, driver, bgi

- 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
- Downloaded from [www.jayaram.com](http://www.jayaram.com) Downloaded from [www.jayaram.com](http://www.jayaram.com) Graphics Mode Graphic function Initialization In order to initialize graphics we `initgraph()` function
- 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
- 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 `graphics.h`
- This variable specifies the graphics driver are applicable only in graphics mode and they communicate directly with monitor
- C offers certain graphics driver and these are the files with BGI extension
- 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
- The `closegraph()` function 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
- For example in a x VGA mode we can address pixels
- If we want to use the graphics mode functions then our first job is to set our computer mode to graphics mode

- 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
- Arun Kumar Yadav Lecturer Eastern College of Engineering Biratnagar Computer Programming in C 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 `int driver 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 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\_154: color, style, syntax, window, line

- Its syntax is `clrscr()` `window()` This function takes four integer arguments that determine the left top right and bottom coordinates of the window
- 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 `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
- 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
- `Int P Drawpoly ( P)` `setlinestyle()` This function can select different style of line
- 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\_155: 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
- xy xy circle( ) The function circle (x y r) draws a circle of radius r The coordinates of the centre of the circle is (x y)
- 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\_156: vertices, fillpoly, polygon, drawpoly, closed

- Arun Kumar Yadav Lecturer Eastern College of Engineering Biratnagar Computer Programming in C 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\_157: 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 ( ) setcolor ( ) Downloaded from wwwjayaramcomnp Downloaded from wwwjayaramcomnp 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 ( ) Er

- Arun Kumar Yadav Lecturer Eastern College of Engineering Biratnagar Computer Programming in C 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