Serial No. Notes Date

Project title :- courier management system

courier management system is a process of organizing last-mile delivery operations. courier management system involves scheduling order deliveries and assigning them to couriers, overseeing operations, planning and optimizing routes, overseeing vehicles, managing fleet, and handling courier expense approvals.

The three objectives of courier management system :-

(i) Deliver products to customers.
(ii) Supports the couriers delivering those products.
(iii) Ensure that customers have a great buying experience, so they come back for more.

Why we need a courier management system?

A courier management system streamlines all the following tasks:

(i) planning and optimizing delivery routes.
(ii) courier tracking and scheduling.
(iii) Analyzing courier performance.
(iv) package tracking during delivery services.
(v) updating customers.

What is the aim of courier management system?

Aim :- Keeping the track of employee information, apart from this details about day to day courier senders, receivers, courier men, hub details and price and product which are sending using courier services.

project description =>

This is a mini project that is written in C programming language.

There are five modules like insert module, display module, search module, delete/update module & sort module etc., that are used in this project.

In this project file handling concept is also used for data storage.

Many other concept like switch case for selection of each module and structure for storing similar data of multiple entities and some other concept like loop (do, for, while), etc are used in this project.

Module explanation =>

(i) Insert Module =>

Insert Module means that, we can add / insert courier details including sender address and receiver address in a file.

(ii) Display module =>

Display module means that, we can display courier details including sender address and receiver address from a file where it is stored by file operators / function methods.

(iii) Search Module :- Search module means that, we can search courier details including sender details and receiver details in a file by using loop concept.

(iv) Delete / update Module :-

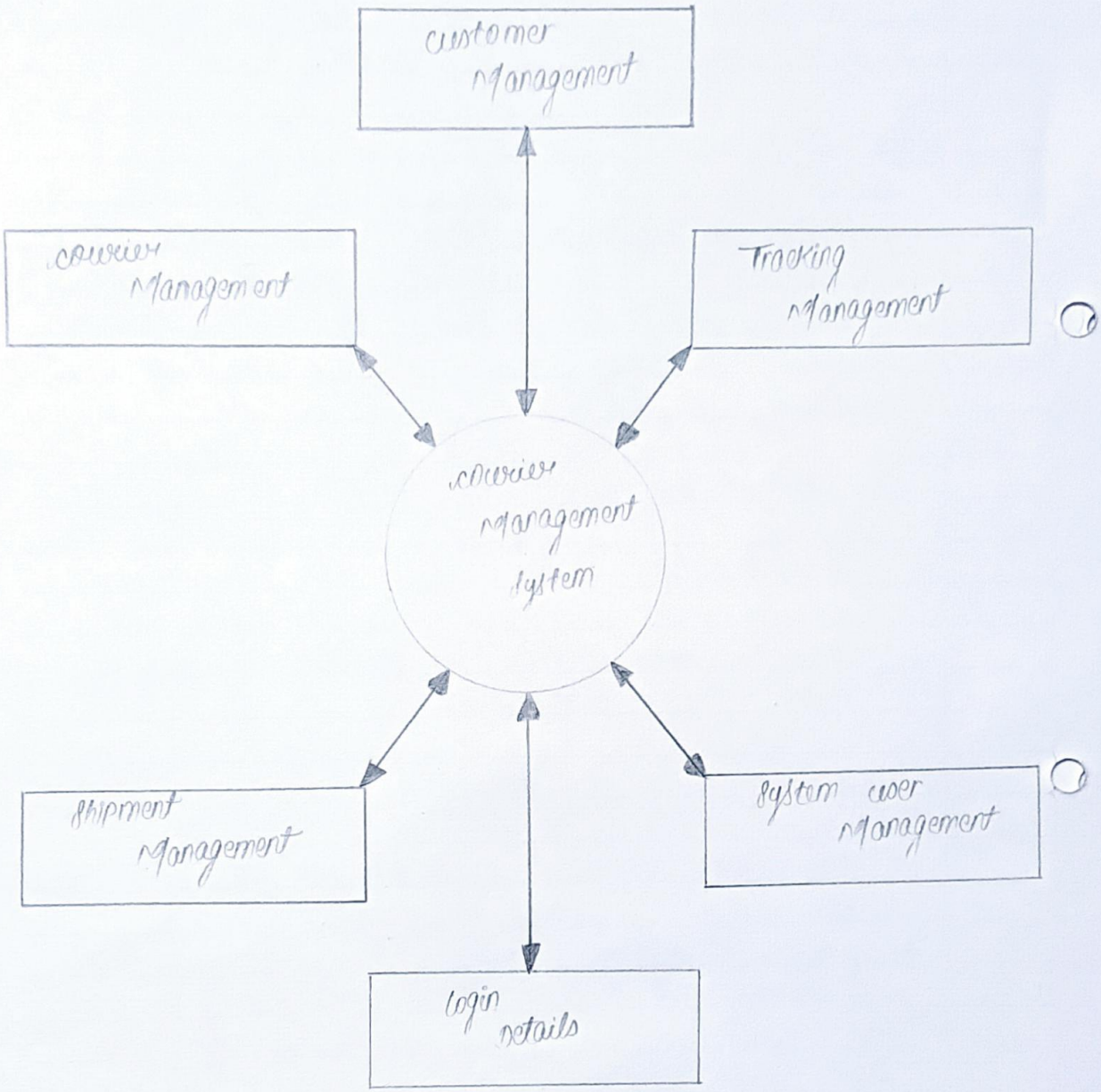
Delete and update Module means that, we can delete courier details including sender details and receiver details or update courier details including sender details and receiver details in a file.

(v) Sort Module :-

Sort Module means that, we can arrange courier details including sender details and receiver details either in ascending or descending order.

What is data flow diagram?

A data flow diagram maps out the flow of information for any process or system. It uses defined symbol like rectangles circles and arrows, plus short text labels, to show data inputs, outputs, storage points and the routes bet" each destination.
Data flow charts can range from simple, even hand-drawn process overviews, to in-depth, multi-level DFDs that dig progressively deeper into how the data is handled.



Zero level DFD = courier management system

Signature

Serial No. Notes Date

Loops in C

In programming language, the purpose of a loop is for executing a block of code repeatedly until the required condition satisfies.

Advantages of loop in C: (i) It provide code reusability.
(ii) We don't have to write the same code again and again.
(iii) It's makes our programming easy.

Types of loop in C: (i) **for loop in C**

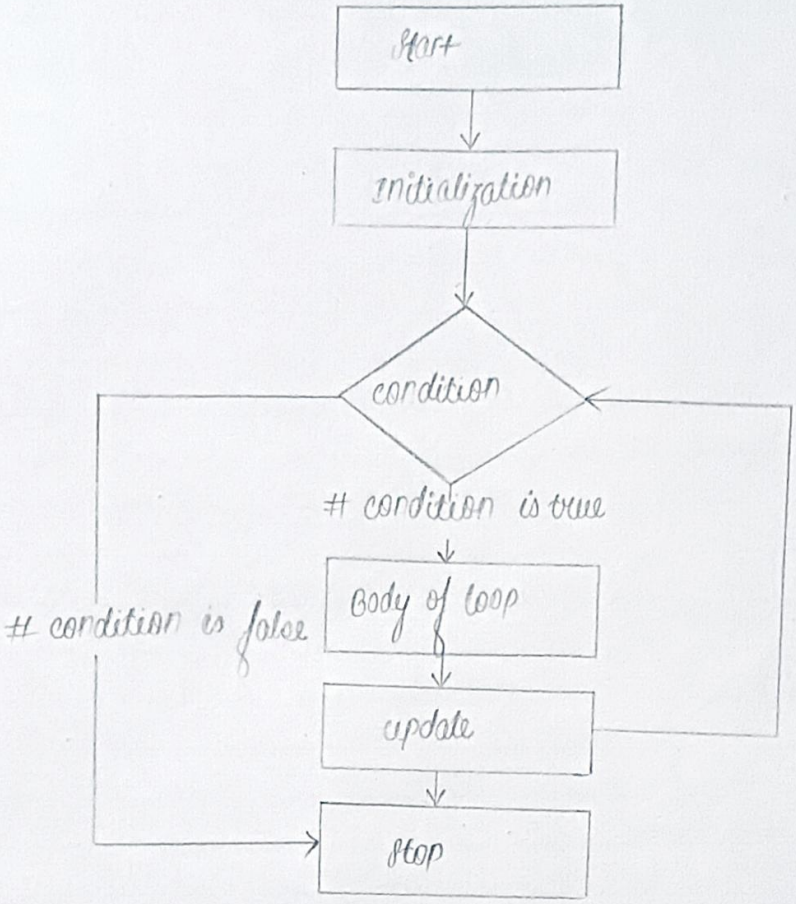
The for loop in C language is used to iterate the statements or a part of program several times. It is frequently used to traverse the data structures like the array and linked list.

Syntax:
```
for (initialization; condition; incr/decr) {
    // statements
}
```

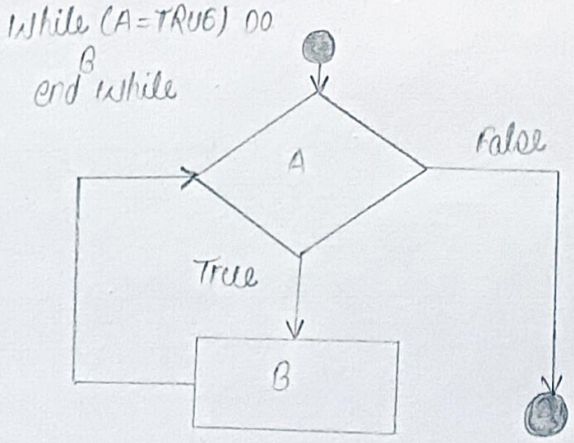
(ii) **While loop in C:** The while loop in C is used to evaluate a test cond^n and iterate over the loop body until the cond^n returns True. The loop ends when the cond^n returns false.
This loop is also known as a pretested loop because it is commonly used when the number of iteration is unknown to the user ahead of time.

Syntax:
```
while (condition) {
    // body of the loop
}
```

Working of for loop =>



working of while loop =>



Signature

(11) Do while loop in C =>

The do/while loop is a variant of the while loop. This loop will execute the code block once, before checking if the condition is true, then it will repeat the loop as long as the condition is true.

Syntax => do {
// body of the loop
} while (condition);

Switch statement in C =>

The switch statement allows us to execute one code block among many alternatives.
We can do the same thing with the if.... else.... if ladder. However, the syntax of the switch statement is much easier to read and write.

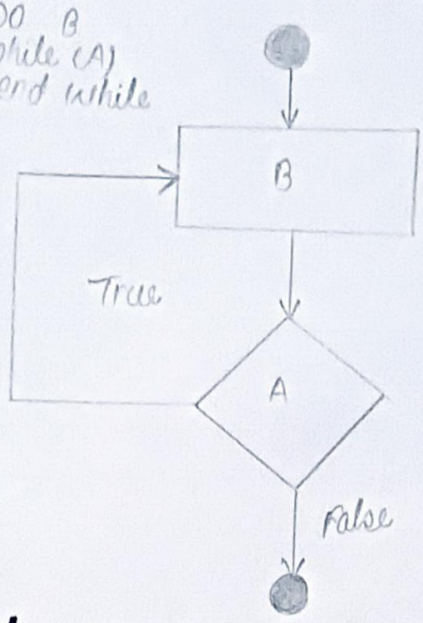
Syntax of switch case =>

```
switch (expression) {
    case constant1 : // statements
        break;
    case constant 2 : // statements
        break;
        :
    default: // default statements
}
```

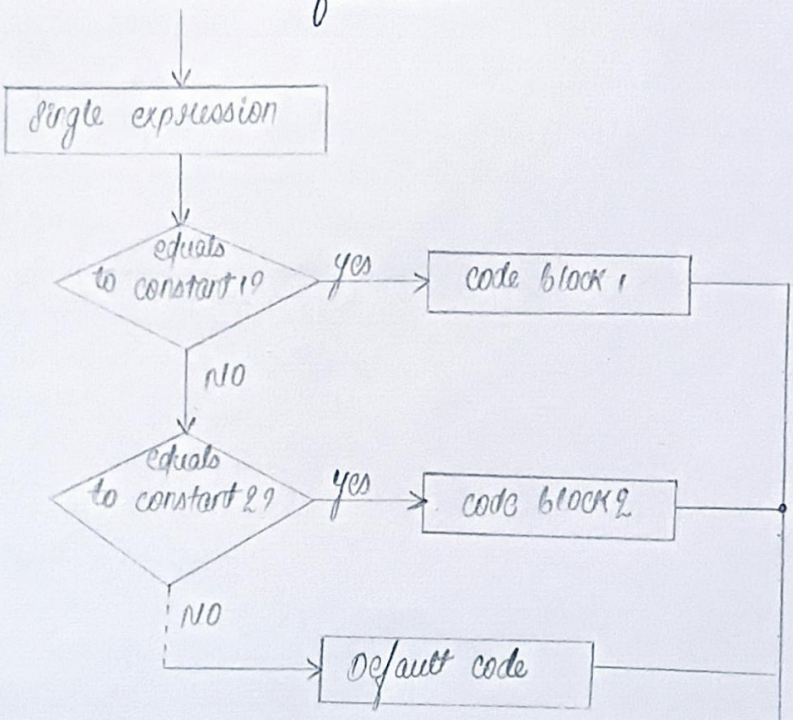
Notes => (i) If we do not use the break statement, all the statements after the matching label are also executed.
(ii) The default clause inside the switch statement is optional.

Working of do while loop

DO B
while (A)
end while



Switch statement flowchart



Signature

C struct (structure) :- In C programming, a struct (or structure) is a collection of variables (can be of different types) under a single name.

To create structure variables, first of all we need to define its data type.

To define a struct, the struct keyword is used.

Syntax :-
```
struct structurename {
    datatype member1;
    datatype member2;
    .....
};
```

Create struct variables :- When a struct type is declared, no storage or memory is allocated. To allocate memory of a given structure type and work with it, we need to create variables.

Syntax :-
```
struct structurename {
    datatype member1;
    datatype member2;
    .....
} variable1, variable2, ....;
```

Access members of a structure :- There are two types of operators used for accessing members of a structure.

(i) . :- Member operator

(ii) -> :- Structure pointer operator.

Keyword typedef :- We use the typedef keyword to create an alias name for data types. It is commonly used with structures to simplify the syntax of declaring variables.

Syntax :-
```
typedef struct structurename {
    datatype member1;
    datatype member2;
    .....
} variable1;
```

Serial No. Date

Notes

C file handling => A file is a container in computer storage devices used for storing data.

Why files are needed => (i) When a program is terminated, the entire data is lost. Storing in a file will preserve your data even if the program terminates.

(ii) We can easily access the contents of the file using a few commands in C.

(iii) We can easily move our data from one computer to another without any changes.

Types of file => (i) Text files.
(ii) Binary files.

(i) **Text file** => Text files are normal .txt files. We can easily create .txt files using any simple text editors such as Notepad. We see all the contents within the file as plain text. They take minimum effort to maintain, are easily readable, and provide least security and takes bigger storage space.

(ii) **Binary files** => Binary files are mostly the .bin files in your computer. Instead of storing data in plain text, they store it in the binary form (0's and 1's). they can hold a higher amount of data, are not readable easily, and provides better security then text files.

File operations => In C, we can perform four major operations on files, either text or binary:

(i) creating a new file.
(ii) opening an existing file.
(iii) closing a file.
(iv) Reading and writing information to a file.

working with files => When working with files, we need to declare a pointer of type file. this declaration is needed for communication b/t^n the file and the program.

Syntax => `FILE *ptr;`

Opening a file - for creation and edit :-
Opening a file is performed using the fopen() function defined in the stdio.h header file.
Syntax :- ptr = fopen("file open", "mode");

Opening modes in standard I/O :-

(i) r :- open for reading. if the file does not exist fopen() returns NULL.
(ii) rb :- open for reading in binary mode. if the file does not exist, fopen() returns NULL.
(iii) w :- open for writing. if the file exists, its contents are overwritten. or, if the file does not exist, it will be created.
(iv) wb :- open for writing in binary mode. if the file exists, its contents are overwritten. or if the file does not exist, it will be created.
(v) a :- open for append. data is added to the end of the file. if the file does not exist, it will be created.
(vi) ab :- open for append in binary mode, data is added to the end of the file. if the file does not exist, it will be created.
(vii) r+ :- open for both reading and writing. if the file does not exist, fopen() returns NULL.
(viii) rb+ :- open for both reading and writing in binary mode. if the file does not exist, fopen() returns NULL.
(ix) w+ :- open for both reading and writing. if the file exists, its contents are overwritten. if the file does not exists, it will be created.
(x) wb+ :- open for both reading and writting in binary mode. if the file exists, its contents are overwritten. if file does not exist, it will be created.
(xi) a+ :- open for both reading and appending.
(xii) ab+ :- open for both reading and appending in binary mode. if the file does not exist, it will be created.

closing a file :- closing a file is performed using the fclose() function.

syntax :- fclose(fptr);

Here, fptr is a file pointer associated with the file to be closed.

Reading and writing to a text file :-

for reading and writing to a text file, we use the functions fprintf() and fscanf().

syntax :- Write to a text file :-

fprintf(fptr, format specifiers, variable);

syntax :- Read from a text file :-

fscanf(fptr, format specifiers, &variable);

Reading and writing in binary file :-

functions fread() and fwrite() are used for reading from and writing to a file on the disk respectively in case of binary files.

syntax :- Writing to a binary file :-

fwrite(addressdata, sizedata, Numberdata, pointer-to-file);

syntax :- Reading from a binary file :-

fread(addressdata, sizedata, numberdata, pointertofile);