Date __/__/____

Inheritance

→ **Base Class:** A class from which another class is inherited.
↳ a.k.a **Super Class**

→ **Abstract Class:** A class which serves only as a base class for other classes and has got no objects. is called an abstract class.

→ **Derived Class:** A class which inherits properties of another class.
↳ aka. Sub-class

→ **Inheritance:** It is the capability of one class to inherit the properties of another class.

→ **Visibility Mode:** The public, private or protected specifiers which control the ~~accessibility~~ visibility and availability of ~~derived~~ inherited. data members in a sub-class.

★ **Important points about Inheritance**

① derived class has all the features of the base class and new features can be added in the derived class

★★ ② The idea of Inheritance implements IS-A relationship.
★

Page No.

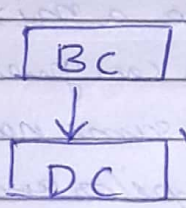
Date ___/___/___

7A Advantages of Inheritance

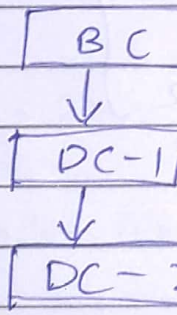
1. Reusability of code.
2. Saves time and effort
3. Easier development and easy to extend.
4. Easier maintenance.
5. Closeness to real-world problems.

Types of Inheritance ☆

1. **Single Inheritance:** Base Class → 1
   derived " → 1 (From)

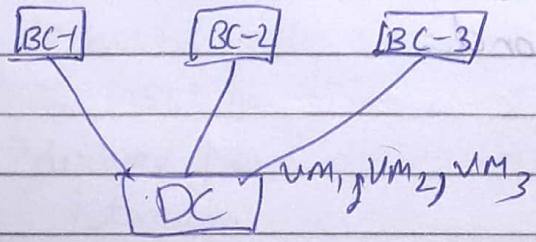


2. **Multilevel Inheritance:** In multilevel inheritance a sub-class inherits from a class which itself inherits from another class.



Base class → 1.
Derived Class → n (count from the first sub-class)

3. **Multiple Inheritance:** a class inherits from multiple base classes, has properties of all of them.



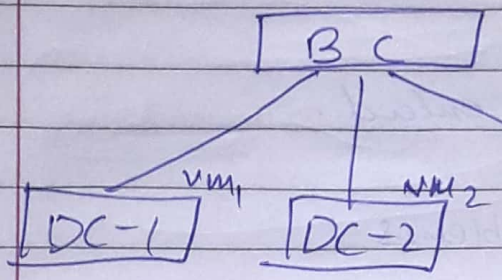
Base Class → n
derived " → 1

Page No.

Date ___/___/___

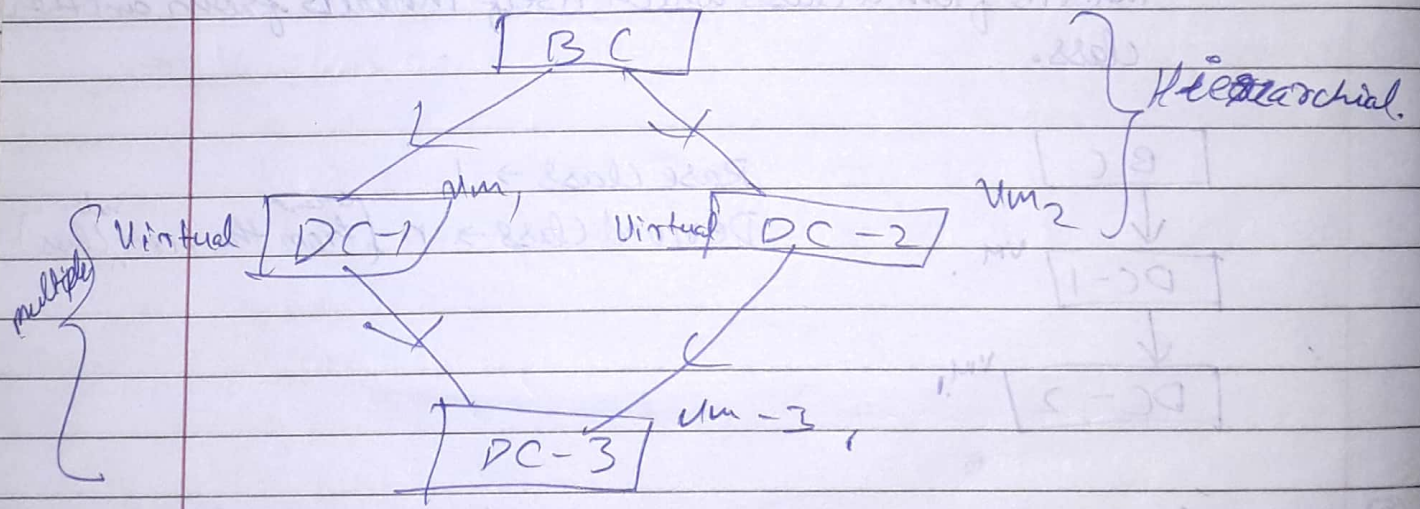
④ Hierarchial Inheritance: multiple classes are derived from same class

Base Class → 1
derived Class → n (from 1 base class)



⑤ Hybrid Inheritance: In this types of inheritance two or more forms of inheritance are combined

* In this type of inheritance we can have a mixture of ~~int~~ number of inheritances but this can generate errors as the use of function with same name from no. of classes confuses the compiler and generates error known as Ambiguity or duciplity, this can be removed by the use of virtual base classes.



Hierarchial - Multiple Inheritance
a.k.a the dreaded diamond.

Page No.

Date ___/___/___

Data file Handling

File: It is a stream of bytes stored on some secondary storage device.

Types of file:

(1) **Text file:** It stores the information in the form of ASCII characters. In text file each line is terminated with a special EOL (end of line) charachter. Some internal translations always takes place when this EOL is read or written.
(.txt)

(2) **Binary file:** It stores the information as it is stored in memory i.e. (binary format). In binary file there is no delimiter for a line also no translation take place as a result binary files are easier and faster to run in program to read and write than text files.
(.dat)

* by default files → text mode.

Stream: It is a general term used to name flow of data i.e. a sequence of bytes

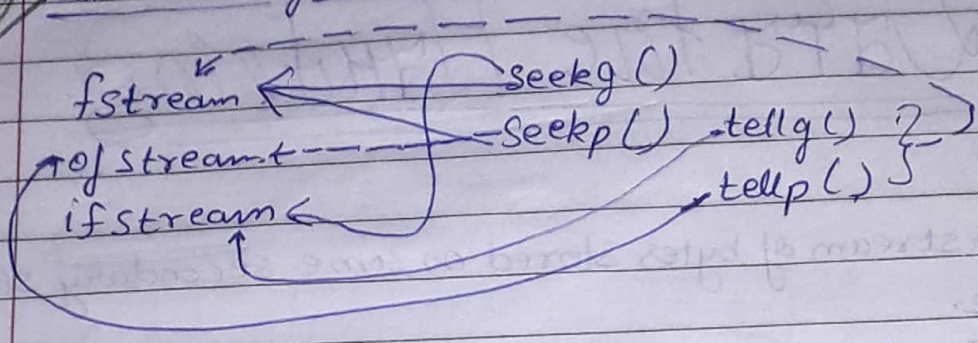
Header file → fstream.h

ifstream	ofstream	fstream
→ fin	→ fout	→ finout
→ read from file.	→ write in file.	→ both read & write
→ ios::in	→ ios::out	→ no default
→ derived from istream	→ derived from ostream.	→ derived from iostream

Page No.

Date ___/___/___

* Pointer Handling in file



seekg()/seekp() → move pointer in file.
tellp()/tellg() → tell the position of pointer.

seekg() → (Byte , offset) 2 parameter.

Byte → file, ~~-ive~~

offset → ios::cur.
ios::beg (default)
ios::end

* difference b/w get() & getline()

Eg.

```
char word[20];
fin.get(word, 20, '$');
```

'$' → On encounter saved in memory.

word → name

20 → how many characters are to be read.

'$' → delimeter character. by default → \n

```
fin.getline(word, 20, '$');
```

'$' → on encounter not saved in memory.

* If open() is not successful, the stream objects store Null (zero) value.
* function close() flushes the buffer before terminating the connection of the file with the stream.

Page No.

Date __/__/__

Buffer : temporary Memory

Binary File Handling

opening & reading

```
ifstream fin;
fin.open("hos.dat", ios::in|ios::binary);
while(fin.read((char*)&h, sizeof(h)))(read everything)
```

Same for writing

```
fin.eof() != 0      EOF reached.
fin.eof() == 0      EOF not reached.
```

for deleting a file.

```
remove("file name.type");
rename("temp file.type", "filename.type");
```

must be same.

Stdio.h .

* file mode :- A constant describing how a file is to be used.

file mode constants are defined in ios class

Page No.