* Member functions :- A member function perform an operation required by the class. It may be use to read, manipulate or display the data member. Member functions of a class can be define in

two place -
inside class definition, outside class definition.

* A program to understand concept of class & object-

```
#include <iostream.h>
#include <conio.h>
class employee
{
public :
int eid;
char name;
};
int main()
{
clrscr();
employee e1, e2;
e1. eid = 1;
e1. name = 'A';
e2. eid = 2;
e2. name = 'B';
cout << e1. eid << "---" << e1. name << "\n";
cout << e2. eid << "---" << e2. name << "\n";
getch();
return 0;
}
```

Output - 1---A
2---B

Program 7:- A program to access the values of private data members.

```
#include<iostream.h>
#include<conio.h>
class sum
{
private: int a,b,c;
public:
void add()
{
clrscr();
cout<<"Enter any two numbers:";
cin>>a>>b;
c=a+b;
cout<<"sum:"<<c;
}
};
void main()
{
sum s;
s.add();
getch();
}
```

Output - Enter any two numbers: 20
30
sum: 50

(A simple program based on encapsulation concept)

* (b) A program of classes and objects -

```
#include <iostream.h>
#include <conio.h>
#include <string.h>
class employee
{
public:
int eid;
char ename[3];
void enterdetails(int id, char name[])
{
eid = id;
strcpy(ename, name);
}
void showdetails()
{
cout<<"Employee ID"<<eid<<"Employee name"
                                <<ename<<endl;
}};
void main()
{
employee e1,e2;
e1.enterdetails(1,"ABC");
e2.enterdetails(2,"XYZ");
e1.showdetails();
e2.showdetails();
cout<<"Employee ID"<<e1.eid<<"---"<<"Employee name"
                                <<e1.ename<<"\n";
cout<<"Employee ID"<<e2.eid<<"---"<<"Employee name"
                                <<e2.ename<<"\n";
getch();
}
```

* Function :- A function is a group of statement that together perform a specific task. Every C++ program has at least one function which is main.

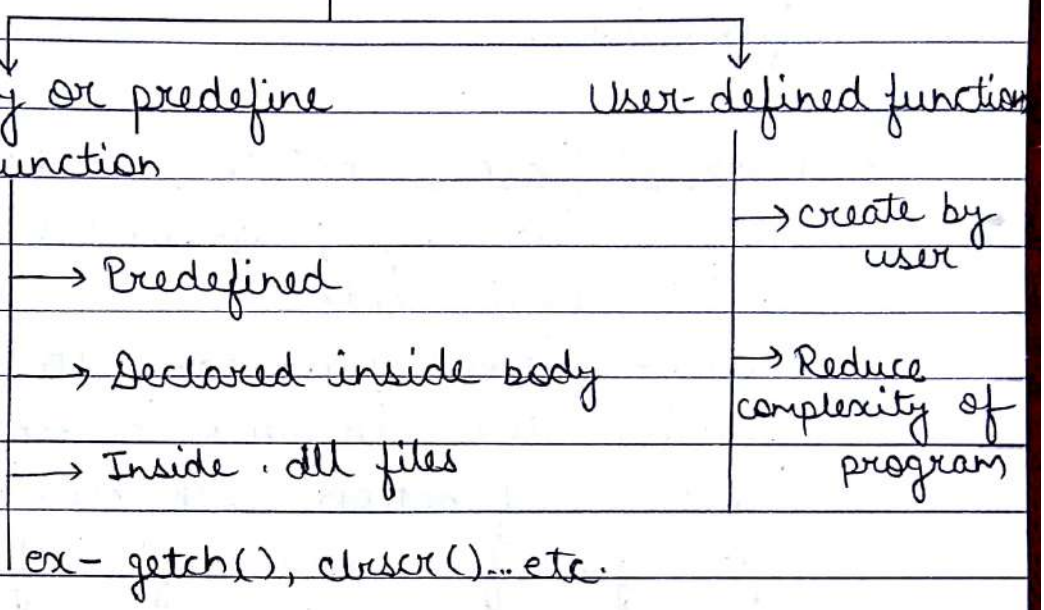
Why use function - Function are used for divide a large code into modules. Due to these we can easily debug and maintain the code.
For ex- If we write a calculator program At the time we can write every logic in a seperate function for addition (sum), subtraction. Any function can be call many time.

* Advantage of function -
1- Code reuse ability.
2- Develop an application in module format.
3- Easy to debug the program.
4- Code optimization, no need to write lot of code.

* Types of function :- There are two types of functions-

1) Library function or pre-define function.
2) User-defined function.

Function



* Library function :- Library functions are those, which are predefine in C++ compiler. The implementation part of predefine function is available in library files that are .lib / .obj files. .lib or .obj files contains predefine code.

Limitations of library functions -

1- All predefine functions are contained limited task only, that is for what porpose function is design for, same purpose it should be use.

2- As a programmer, we do not have any control on predefine functions. Implementation part is there machine readable format.

3- In implementation whenever a predefine function is not supporting the user

requirement then go for user define functions.

* User-defined functions – These functions are created by programmers according to their requirements.
For ex- Suppose you want to create a function for add two numbers then you create a function with name sum. This type go for user define functions.

* Defining a function – Defining of function is nothing but give body to functions that means write logic inside function body.

1- Return type – A function may return a value. The return type is the value the function returns. Return type parameters and return statement are optional.

2- Function name – A function declaration is the process of tell the compiler about a function return.

3- Parameters – A parameter is like a place-holder. When a function is invoke you pass a value to the

parameters. This value is refer to the type ordered and number of the parameter of a function. Parameters are optional that is a function may contain number of parameters.

4- Function body – The function body contains a collection of statements that define what the function does.

5- Function Declaration – Function declaration tells the compiler about a function name and how to call the function. The actual body of the function can be define seperately.
Syntax:

```
return_type functionname (Parameters);
```

6- Calling a function – When we call any function control goes to function body and execute entire code. For call any function just write name of function and if any parameter is required when pass parameter.

(8) For example:

```
#include <iostream.h>
#include <conio.h>
void sum();
```

```
int a = 10, b = 20, c;
void sum()
{
c = a + b;
cout << " sum:" << c;
}
void main()
{
clrscr();
sum();
getch();
}
```

Output :- sum : 30

→ calling function