Assignment - 15

Q1] What is the concept of function overloading?

→ In case of function overloading we can define multiple function in a class with same name & different parameter

- function overloading is a feature of object oriented programming where two or more functions can have the same name but different parameters.

- In function overloading function name should be the same and the arguments should be different.

Q2] Explain the different between constant function and non constant function

constant function	non-constant function
1) constant function is declared with the const keyword	1) A non-constant function can only be called by non constant object.
2) A constant function cannot modify any non-static members	2) A non-constant function can modify the object it is called on

3) Syntax: function name () const { }	3) Syntax: function name () { }
4) A const member function can be called any type of object	4) Non-const functions can be called by non-const objects only

Q3] Explain the concept of member initalization list of constructor in C++

→

Using initalization lists to initalize data members of constructor can be convenient if you don't need to do any error-checking on the ~~or~~ constructor arguments.

- in c++ where the use of an initalize a data member is actually required

1) Data members that are const but not static must be initalized using an initalization list.

2) Data members that are reference must be initalized using an initalization list

3) An initialization list can be used to explicitly call a constructor that ans takes arguments for a data member is that is an object of another class.

4) In Derived class constructor, an initialization list can be used to explicitly call a ~~ctes~~ base class constructor that. takes arguments.

(4) Can we use the pointer in case of static member function of class

→ - You cannot use a pointer to member to point to a static class member in c++

- You can use a normal pointer to point to a static class member.

- A static member function is not associated with any object.

- A static member function does not have a this pointer.

- Pointers to member functions are used to refer to nonstatic members of class objects.

Q5) How can we initalize the constant characteristics of class

→

As a c++ is oop language the concept of constant is applicable for characteristic and behaviours.

- In case of constant characteristics in which we cannot change the values of its ~~if~~ ones they decla red as constant.

Syntax:

```
const data-type character-name = value;
```

- To initalize the const value using constructor; we have to use the initalize list.

- this initalizer list used to initalize the data members of class

Q6] Explain what are the limitations of static function of a class

→ To call the static function we use the name of that class

- static function can be call without creating object

- static function can Access only static characteristics of class

- static functions can't be declared as virtual functions

- static functions can't be inherited or overridden

Q7] How to initalize the static constant characteristics of a class?

→

To initialize the static constant characteristics of a class in C++ you can use the const keyword in the declaration & initialization.

- initializing static const characteristic.

```
static const int i = 14;
```

Q8) Explain why constant object can not call constant member function of class?

→
- Constant object is a such object who all the characteristics are constant
- like a normal object we can create constant object
- In case of constant object all the characteristic of that object are consider as constant

Q9] Can we call static function of a class using object of a class?

→
- Yes you can call a static function of a class using an object of the class but its not recommended
- The recommended way to call a static function is to use the scope resolution operator (::)

e10] what is difference between constand input argument and non constand arguments of function with program

→ constant input arguments

In C++ constant argument in function indicates that the values passed to the function as parameters cannot be modified inside the function

for example:-

```
#include <iostream>
using namespace std;
int main ()
{
void display (const int i=10)
{
    i++ // not allowed
printf (" value of i ")
cout << " value of i" << endl;

return 0;
}
```

Non constant Arguments

- In C++ Non constant Arguments of function means normal parameters of function will be modified

for example:

```
#include <iostream>
using namespace std;

int main ()
{
    void display (int i)
    {
        i++;
    }

return 0;
}
```