

Assignment No. - 15

1. what is use of function overloading?

-
- ° Function overloading is the type of compile time polymorphism.
 - ° In function overloading we overload or load the function with different parameter & different datatype.
 - ° In overloading, we define same functions name with [class name] same name with different prototype.
 - ° overloading in programming is considered as syntactic sugar.
 - ° The use of overloading is for defining multiple methods with same name and different prototype, for code simplification.

° for eg. -

```
class Add {
```

```
    public:
```

```
        // int i, j, k;
```

```
        int Add (int a, int b)
```

```
{
```

```
    int c = 0;
```

```
    return c = a + b;
```

```
}
```

```
int Add ( int a, int b, int c )  
{  
    int c = 0;  
    return c = a + b + c;  
}
```

};

```
int main()
```

{

```
    int Ans = 0;
```

```
    // Add obj (10, 20);
```

```
    // Add obj2 (10, 20, 30);
```

```
    Add obj;
```

```
    Ans = obj.Add (10, 20);
```

```
    cout << "Addition is : " << Ans << endl;
```

```
    Ans = obj.Add (10, 20, 30)
```

```
    cout << "Addition is : " << Ans << endl;
```

```
    return 0;
```

}

2. what is difference between constant function and non-constant function in c++?

→ 1. Constant Function: It is a type of function which cannot change the value inside it.

• for eg.

```
int i = 10; int j = 20;

void fun() const
{
    i++;
    j++;
}
```

} // Not allowed

2. Non-const Function: It is a regular function in which we can change the value of function-variables.

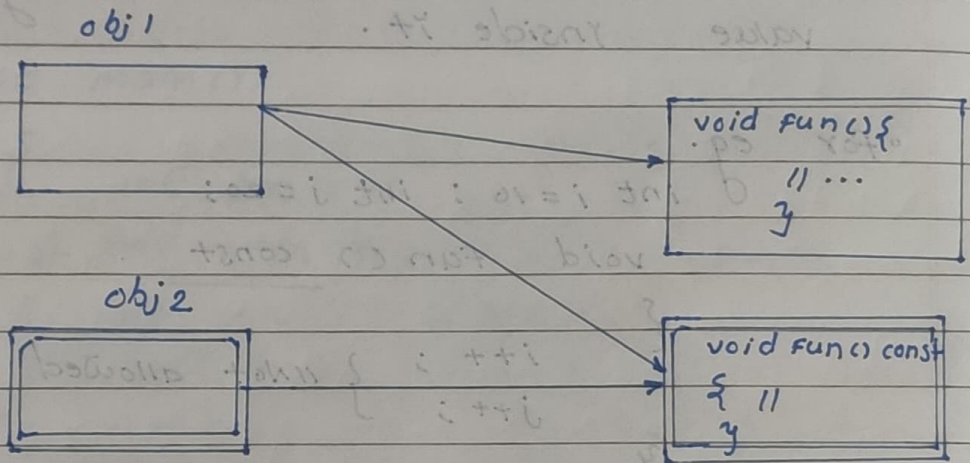
```
void fun()
{
    i++;
    j++;
}
```

} // Allowed

Note

1. constant function - This function can be called by using both constant & non-constant object created.

- Non-const function - This function can be called using only non-constant object of class.



Q.3 what is mean by member Initialization
list constructor in c++?

- When we create the constant characteristics in the class then we can initialize those characteristics without the constructor also but non-constant characteristics should be initialized using constructor.

• for eg. - class Demo
 {
 public:
 const int i = 10;
 const int j = 20;
 };

• Here, problem is that we cannot take value from user here i, j so by using constructor initialization list we can take input for i, j any constant characteristics from user i.e

```
class Demo
```

```
{
```

```
public
```

```
const int i;
```

```
const int j;
```

Initialization
List

```
Demo (int a, int b) : i(a), j(b)
```

```
{
```

```
//
```

```
}
```

```
};
```

Q.4 How to initialize constant characteristics of class?

→ • We can initialize constant characteristics in class using const keyword.

• We can't increment and decrement the value of constant characteristics.

```
class Demo { public:
```

```
const int a = 10;
```

```
const int b = 20;
```

```
void func() {
```

```
a++; // NA
```

```
b++; // NA
```

```
};
```

```
}
```

Q.5 can we use the concept of this pointer in case of static member function of class?

→ • No, we cannot use the concept of this pointer in case of static member function of class.

• This pointer takes first hidden implicit parameter i.e objects address only and this pointer can be used for only non-static.

• static member functions can be called without any object (with class name).

Q.6 Wrong can we call static member function of a class using object of a class.

→ • we cannot call static member function of a class using an object of a class.

• static member function belongs to class itself.

• To call static member function we need :: operator & name.

• for e.g class Demo {

public :
static void demo { }

}
};

- we can call above member function using class name & :: scope resolution operator as

Demo:: Demo1();

or demo obj; obj.Demo1();

Q.7 what are the limitations of static function of class?

→ 1. we cannot access non-static datamembers of a class, static function can be called by only static function object.

2. cannot be overridden by derived class.

- cannot inherit static function.

3. Static functions only gets memory for only once, not other functions.

i.e no. of obj = like new memory for each.

Q.8 How to initialize static constant characteristics of class?

→ The static constant characteristics of class can be initialized at the time of definition only, initialization list can give error.

e.g - static const int k = 10;

Q.9 why constant object cannot call non-constant member function of a class.

→ • constant object is a such a object whose all the characteristics are considered as constant.

• constant object can call only constant function.

• But, in case of non-constant object of class can call both methods i.e static as well as non-constant.

• In case of const object all characteristics of that object are considered as constant.

• According to rules of c++ non-const object can call both and const object can call only const function.

Q.10 what is difference between constant input argument & non-constant input argument of function.

→ • Const Argument - It is a type of the const keyword is used and argument cannot change their value.

• Non-const Argument - In this type of argument value of argument can change their value as per given argument.