LAB 07

Das Anjan Kumar

D42DQA

b) Examine the operator= and answer the following questions:

- Under what constraints can we call operator= from a copy constructor?

We can call operator= like this in the copy constructor,

```
Vector::Vector(const Vector& theOther) { //copy constructor

        *this = theOther;
}
```

And this will work like the copy constructor, but we have to aware, because function of both are not same. As we know copy constructor is called when a new object is created from an existing object, as a copy of the existing object. And assignment operator is called when an already initialized object is assigned a new value from another existing object.

- Why are both parameter and returning value passed by reference?

Both parameter and returning value are passed by reference because we want to make sure that we make the changes by address, so it takes place everywhere, noy just locally. Also It allows us to have the function change the value of the argument. Return by reference is also fast, which can be useful when returning classes.

- What happens with self-assignment like v=v?

In our case nothing happens in case of self assignment, because our operator overloading is handled in a efficient way. But in some case the program results in unpredictable behavior.

c) What modifications need to be done at Vector class to be able to manipulate userdefined data types instead of a built-in int type (e.g. to have a vector of String objects)?

- We just need to include the user-defined data type in the vector class to use. Let's demonstrate with an example. Suppose we have a user defined class Newclass,

```
class Tera {

        int x;
        Tera() {
                x = 0;

        }

};
```

```
We just need to declare this class in Vector class, like this,
```

```
class Vector
{
        ………………………
        unsigned int elementNum;
        int* pData;

        Tera x;
        ……………

}
```