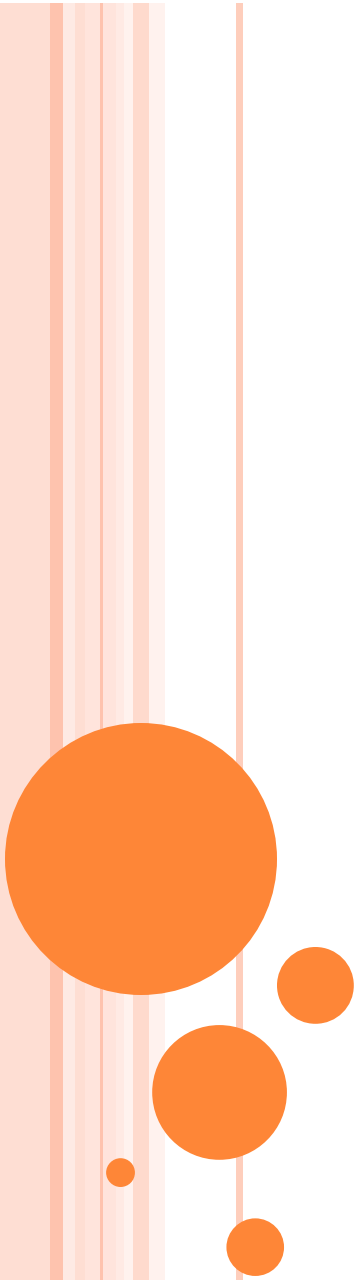

DArray Class

CS A250 – C++ Programming II

A Pointer as a Member Variable

- Having a **pointer** as a **member variable of a class** means that the pointer will be pointing at a **dynamic variable** that is stored in the **heap**
  - The dynamic variable is **not** *physically* part of the object
  - BUT the pointer is part of the object
- We have already seen this when implementing linked lists.

A Pointer as a Member Variable (cont.)

- The class **DArray** creates objects that contain three variables:
  - An **int** to store the **capacity** of the array
  - An **int** to store the **number of elements** in the array
  - A **pointer** that will **point** to an **array of integers**

CLASS DARRAY

```
const int CAP = 100;

class DArray
{
public:
    DArray( );
    // other functions
    ~DArray( );

private:
    int *a;      //will point to an array of integers
    int capacity;
    int noOfElem;
};
```

```
// default constructor

DArray::DArray()
{
    capacity = CAP;
    noOfElem = 0;

    a = new int[capacity];
}
```

Class DArray (cont.)

Object of the class DArray

```
int * a = [array address]
int capacity = 10
int noOfElem = 6
```

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
35	67	91	42	73	15				

[array]

Class Darray (cont.)

Object of the class DArray

```
int * a = [array address]
int capacity = 10
int noOfElem = 6
```

What does that mean?

It means that you need to think carefully when adding a **const** modifier to a **member function** of the class **Darray.**

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
35	67	91	42	73	15				

[array]

The `const` Modifier on Functions

- **Recall:**
  - You add a `const` to a **member function** when the member variables of the class will **not** be modified.

THE const MODIFIER ON FUNCTIONS (CONT.)

Object of the class DArray

```
int * a = [array address]
int capacity = 10
int noOfElem = 6
```

Should a member function that replaces one element in the array be **const**?

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
35	67	91	42 81	73	15				

[array]

THE const MODIFIER ON FUNCTIONS (CONT.)

Object of the class DArray

```
int * a = [array address]
int capacity = 10
int noOfElem = 6
```

Should a member function that replaces one element in the array be **const**?

Yes! The member variables of the object will not be modified.

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
35	67	91	42 81	73	15				

[array]

42 was simply replaced by 81.

THE const MODIFIER ON FUNCTIONS (CONT.)

Object of the class DArray

```
int * a = [array address]
int capacity = 10
int noOfElem = 6
```

Should a member function that deletes an element in the array be **const**?

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
35	67	91	81	73	15				

[array]

THE `const` MODIFIER ON FUNCTIONS (CONT.)

Object of the class DArray

```
int * a = [array address]
int capacity = 10
int noOfElem = 6  5
```

Should a member function that deletes an element in the array be **const**?

No. The number of elements will be decremented, modifying the member variable of the class.

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
35	67	91	81 73	73 15	15				

[array]

81 was deleted by shifting all elements to the right of 81.
The number of elements in the array is now 5.

THE const MODIFIER ON FUNCTIONS (CONT.)

Object of the class DArray

```
int * a = [address of array]
int capacity = 10
int noOfElem = 6
```

Do **not** forget that **a** is a pointer!

```
bool Darray::compareElemAtIdx (const Darray& otherArray,
                                                int idx) const
{
      return (a[idx] == otherArray.a[idx];
}
```

This pointer is referring to the **calling object**.

This pointer is referring to the object **otherArray**.

The DArray Class

- **Project:** DArray class

DArray Class (end)