

# Object Oriented Programming

## Classes continued ...

Mr. Usman Wajid

*usman.wajid@nu.edu.pk*



**National University**  
of Computer & Emerging Sciences

# Destructor

## Destructor

It is a function same as the class name, stars with ~ symbol, without any type. The destructor is automatically executed when the class object goes out of scope

- A class can have only one destructor
- Example,

```
class Student {  
    public:  
    Student() {  
        cout << "Welcome!\n";  
    }  
    ~Student() {  
        cout << ("Good bye\n");  
    }  
};
```

# Destructor Example

```
class Student {  
    public:  
    Student(){  
        cout<<("Welcome\n");  
    }  
    ~Student(){  
        cout<<("Good bye\n");  
    }  
};  
  
void test(){  
    Student mahad;  
    cout<<"exiting test()\n";  
}  
  
int main() {  
    Student ali;  
    test();  
    test();  
    cout<<"exiting main()\n";  
}
```

# Destructor Example

```
class Student {  
    public:  
    Student(){  
        cout<<("Welcome\n");  
    }  
    ~Student(){  
        cout<<("Good bye\n");  
    }  
};  
  
void test(){  
    Student mahad;  
    cout<<"exiting test()\n";  
}  
  
int main() {  
    Student ali;  
    test();  
    test();  
    cout<<"exiting main()\n";  
}
```

```
Welcome  
Welcome  
exiting test()  
Good bye  
Welcome  
exiting test()  
Good bye  
exiting main()  
Good bye
```

# Array of Class objects

- We can create an array of class objects
- Syntax,

```
int main () {  
    <class-name> <object-name>[100];  
}
```

# Array of Class objects: practice

- Write a program that depict the following Memory structure,
- whereas, the class must have a default constructor that assign a default value, i.e, 0 to the hr, min and sec variables
- The constructor, must use the default argument values to pass values to the private variables using the set\_functions

