

# Lab 11

# Validating Input

What are you going to validate in the context of this week's exercise?

- All inputs from the user that contains an **int** value:
  - Menu Selections
  - Ages
  - Drivers license numbers

# Validating Input - Important Functions

- **cin.fail ( )**: This function returns true when an input failure occurs. In this case it would be an input that is not an integer (or whatever the type of the input variable that you have specified). If the cin fails then the input buffer is kept in an error state
- **cin.clear ( )**: This is used to clear the error state of the buffer so that further processing of input can take place
- **cin.ignore ( )**: This function is used to ignore the rest of the line after the first instance of error that has occurred and it skips to or moves to the next line
- **cin.eof ( )**: This function can be used to check end of file errors. This returns 1 if the program tried reading something but, it was from the end of the file

# Destructors

- They are usually used to deallocate memory and do other cleanup for a class object and its class members when the object is destroyed
- It has the same class name
- It doesn't have any arguments
- It has no return type
- A destructor is called when: a function ends, a program ends, a block containing local variables ends, a delete operator is called.

# Destructors

- Same class name
- Preceded by a tilde (~)

# Exercise

Read data from file and allow the user to interact with this data and print to the screen as per the user's choice

55

381043 Francine Palau 23 N

449122 Kam Swindler 57 Y

401934 Migdalia Constable 21 Y

# Exercise

- You'll have 2 classes: a **DriversLicenseRecord** class and a **DMV** class

## **DriversLicenseRecord**

- **First name**
- **Last name**
- **Age**
- **Voter Status**

## **DMV**

- **Constructor**
- **Destructor**
- **print menu**
- **Validate user input**

## Exercise: main.cpp

```
int main(int argc, char** argv)
{
    //Do a check to make sure we have the right number of
arguments,
    //exit if there aren't enough arguments

    std::string fileName = ???; //get the file name from argv
    DMV myDMV(fileName);
    myDMV.run();
}
```