Agenda

- Exception Handling
- Association
- Manipulators

Exception Handling (demo01 and demo02)

- When we provide any wrong input to the program then the program may generate some errors or it may get crash.
- to avoid such errors or crashing of program we need to handle these errors.
- to handle these errors we use exception handling.
- to perform exception handling we use below 3 keywords

1. try
    - It is used to look for the statements wether they are generating the exception or no.
    - If exception is generated it will look for the matching catch block and will execute that catch block.
    - If matching catch block is not found then the program will terminate by throwing the type of exception generated.
    - Every try block should have atleast 1 catch block
2. catch
    - It is block which is used to handle the exception.
    - we can write multiple catch blocks for a single try block.
    - if we want to handle multiple exceptions inside single catch block we can write a generic catch block.
    - If you want to keep normal and genric catch block then the generic catch block shoud be the last catch block of the series.
3. throw
    - It is used to generate an exception

Association (has-a) (demo03)

- When has-a relationship exists between two entities/classes we use association
- eg
    - Car has-a Engine
    - Car has-a AudioSystem
- It is further classified into two types

1. Composition
    - If the entities are tightly coupled we use Composition
2. Aggegration
    - If the entities are loosely coupled we use Aggegration

Manipulators (demo04)

- It is used to manipulate the output i.e to only change how the output looks and not the actual value.
- Two types of manipulators

1. Without arguments (Parameterless)
2. With arguments (Parameterized)