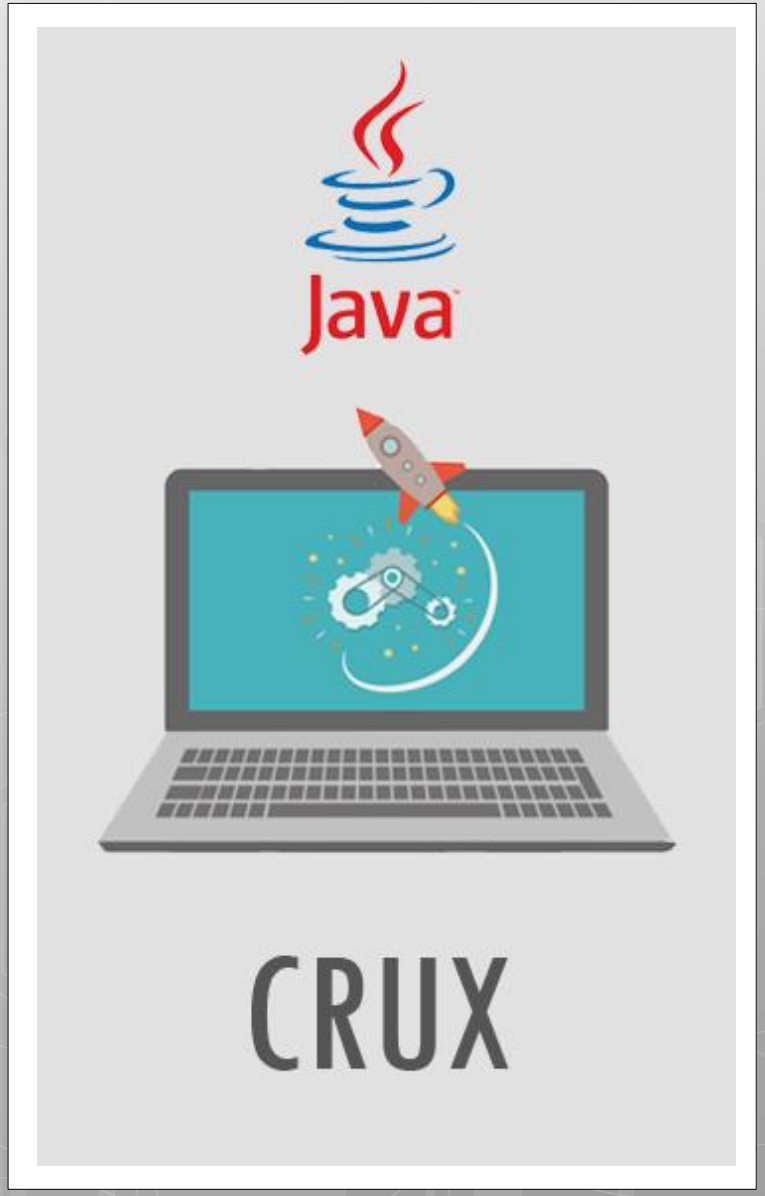


CODING BLOCKS

OOPs 2

- Inheritance
- Polymorphism
- Stacks and Queues

Sumeet Malik

Inheritance

- Reusability (of code and signature)
- Extending functionality of an existing class
- Add new methods and data members to the derived class
- Override existing functions of base class in derived class.

Polymorphism

- Maintainability
- Same name, many forms
- Compile Time
  - ~~Operator Overloading~~
  - Function Overloading
  - Generics
- Runtime
  - Virtual functions

CODING BLOCKS

Compile Time Polymorphism

- Operator Overloading
  - Not allowed in Java
- Function Overloading
  - Many functions of same name but varying signature in the same class
  - Return type not part of signature
  - Varargs?
- Generics - Later

Runtime Polymorphism

- Context
  - P – parent class
  - C – child class
- P obj = new P();
- P obj = new C();
- ~~C obj = new P();~~
- C obj = new C();

Runtime Polymorphism

- Compiler has its eyes on the LHS or the reference.
  - Compiler will compile code congruent to the LHS.
- JVM has its eyes on the RHS or the object that got created.
  - JVM will invoke functionality congruent to the RHS.

CODING BLOCKS

Exceptions to Rule

- Data members
- Static Functions

Runtime Polymorphism

- Overriding base class functions
  - All functions in Java are by default virtual, hence over-rideable
- Variables
- Collections
- Functions accepting parameters
- Final, Abstract, Static implications?

CODING BLOCKS

Final Classes and Functions

Abstract Classes and Functions

Data members modifiers

- Public
- Protected
- Private
- Nothing(Friendly)
- Final
- Static

Function modifiers

- Public
- Protected
- Private
- Nothing(Friendly)
- Abstract
- Final
- Static

Class modifiers

- Public
- Private
- Nothing(Friendly)
- Abstract
- Final

Dynamic Stack

CODING BLOCKS

Stack Class

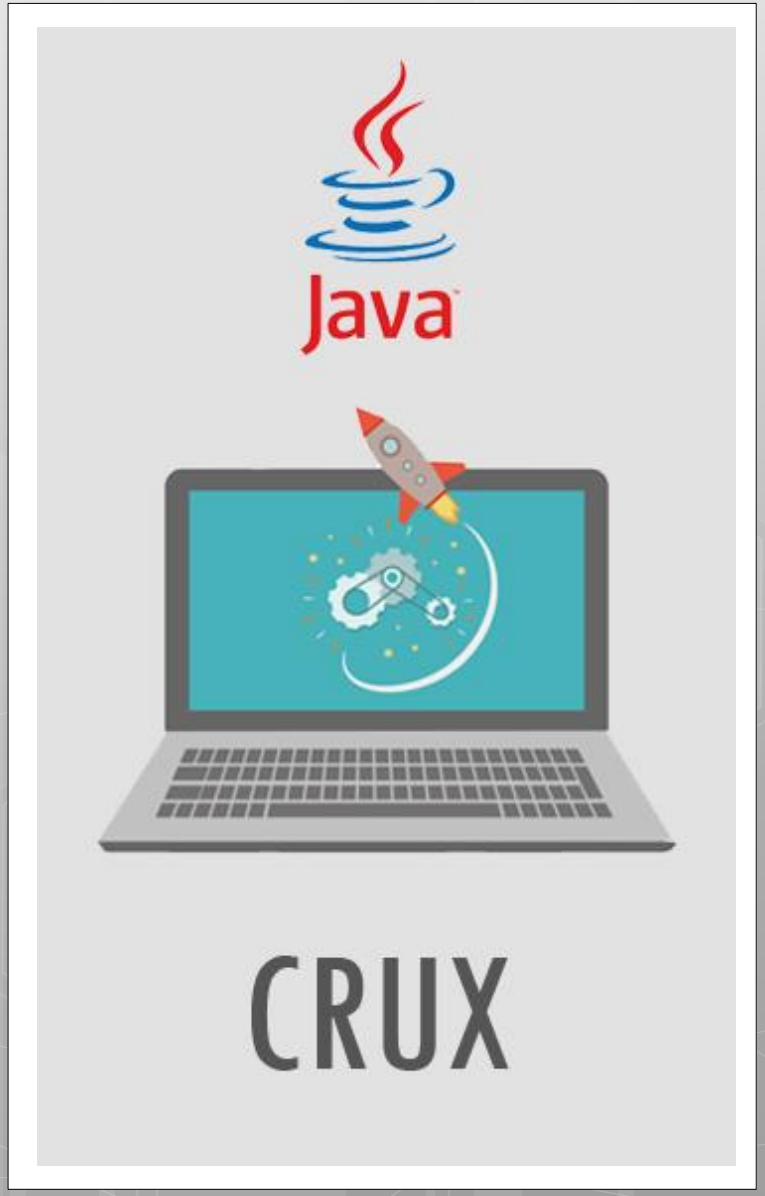

```
public class DynamicStack extends Stack {
        public void push(int item) throws
        Exception;
}
```

Dynamic Queue

CODING BLOCKS

Queue Class


```
public class DynamicQueue extends Queue {
      public void enqueue(int item) throws
      Exception;
}
```



CODING BLOCKS

Thank you

Sumeet Malik