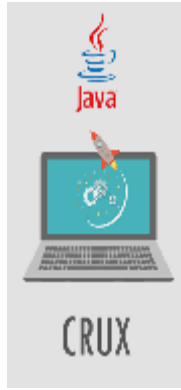
CODING BLOCKS

OOPs 1

- Encapsulation
- Stacks
- Queues

Sumeet Malik



Object Oriented Programming

CODING BLOCKS

Programming Languages

- Helps the programmer translate his ideas in natural language to something a machine can understand
- Natural Languages - Verbs, Nouns, Adjectives, Adverbs

CODING BLOCKS

Programming Languages

- Functional – Focus on verbs
- Object Oriented – Focus on nouns (verbs and adjectives are treated as something related to nouns)

OOP vs FP

- OOPs provides better modularity
  - Abstraction
  - Data Hiding
- OOPs provides better reusability
  - Inheritance
- OOPs provides better maintainability
  - Polymorphism

CODING BLOCKS

Java – Object Oriented

- Classes and Objects
- Data
- Functions

Classes and Objects

- Classes are blueprints to create objects
- Objects are the individual instances created using classes
- Copy of only non static data members is created

CODING BLOCKS

Data Members

- Static vs Non static
- Final
- Initialization
  - new
  - this
  - Parsing
  - Constructor

CODING BLOCKS

Constructors

- Default
- Parametrized
- Copy Constructor
- Assignment Operator
- Destructor – GC, Finalize, Mark and Sweep

CODING BLOCKS

Static vs Non-static Functions

CODING BLOCKS

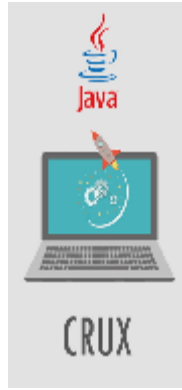
Pillars of OOPs

- Encapsulation
- Inheritance
- Polymorphism

Encapsulation

- Modularity
- Bind the data and functions together
  - Classes and Objects
- Make the state safe
  - Data Hiding
- Hiding the implementation details
  - Abstraction

CODING BLOCKS



Encapsulation – Data Hiding

Public vs Private

CODING BLOCKS

Encapsulation - Abstraction

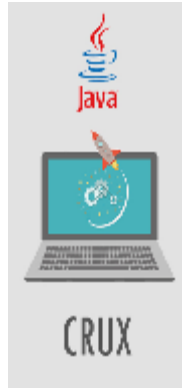
- To use the class all you need to know is public API of the class
  - Public Functions
  - Input format
  - Output format
- Ignorance is bliss
  - Well, ignorance from unnecessary inner details.
  - Ignorance from private functions

CODING BLOCKS

Stacks

Stack Class


```
public class Stack {
        public int size();

        public boolean isEmpty();

        public void push(int item) throws  Exception;

        public int pop() throws Exception;

        public int top() throws Exception;

        public void display();
}
```



Queues

Queue Class

```
public class Queue {
        public int size();

        public boolean isEmpty();

        public void enqueue(int item) throws
        Exception;

        public int dequeue() throws Exception;

        public int front() throws Exception;

        public void display();
}
```

CODING BLOCKS

Thank you

Sumeet Malik