31/07/15

Crux

Lecture -9

Object Oriented Programming-2

Manisha Khattar

Object Oriented Programming

Encapsulation

1. Bind the data and functions together
2. Hiding the implementation details
3. Lets us change the implementation without breaking code of our users

CODING BLOCKS

Inheritance

1. Extending Functionality of an existing class
2. Add new methods and fields to derived class
3. If both classes have a function with same name, which class's function will get called?

CODING BLOCKS

Inheritance

1. Super
2. How constructors are called ?

CODING BLOCKS

Polymorphism

1. Overriding the base class functions(Virtual Functions)
2. Ability of a variable to take different forms
3. Ability of a function to behave differently on basis of different parameters
4. Ability of a function to work with parameters of subtypes

CODING BLOCKS

Final Class?

Final Function?

Abstract functions (Pure Virtual)

Abstract Classes

Data Member Modifiers

1. Public?
2. Protected?
3. Private?
4. Nothing(Friendly)
5. Final
6. Static

CODING BLOCKS

Function Modifiers

1. Public?
2. Protected?
3. Private?
4. Nothing(Friendly)
5. Abstract
6. Final
7. Static

CODING BLOCKS

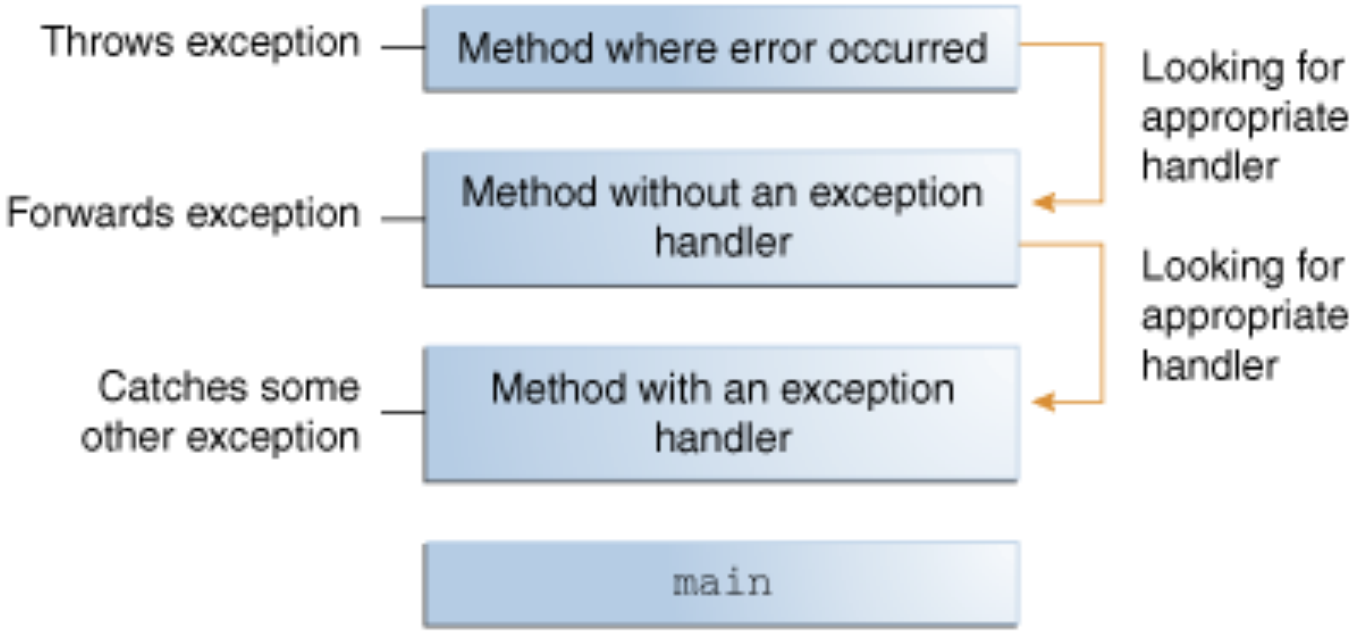
Classes Modifiers

1. Public?
2. Nothing(Friendly)
3. Abstract
4. Final

CODING BLOCKS

Exceptions

Exceptions & the call stack



CODING BLOCKS

Type of Exceptions

1. Checked Exceptions (java.lang.Exception)
2. Errors(java.lang.Error)
3. Runtime Exceptions (java.lang.RuntimeException)

CODING BLOCKS

How to throw Exceptions?

Either Catch or Specify

How to create our own Exception Class?

CODING BLOCKS

Try catch and finally?

Throwable?

Thank You!

Manisha Khattar

manisha@codingblocks.com