

CURSO JAVA BASICO

OBJETIVO

Que el estudiante entienda y aplique las funciones, construcciones, librerías y metodologías orientadas a objetos del lenguaje de programación Java

TEMARIO

- **Breaking the Surface**

- The Way Java Works
- What you'll do in Java
- A very brief history of Java
- Sharpen your pencil
- Sharpen your pencil answers
- Code structure in Java
- Anatomy of a class
- Writing a class with a main
- What can you say in the main method?
- Looping and looping and...
- there are no Dumb Questions
- Conditional branching
- Coding a Serious Business Application

- **Know Your Variables**

- Declaring a variable
- "I'd like a double mocha, no, make it an int."
- You really don't want to spill that...
- Back away from that keyword!
- This table reserved.
- Controlling your Dog object
- An object reference is just another

- Monday morning at Bob's
- Phrase-O-Matic
- Fireside Chats
- Exercise: Code Magnets
- Exercise: BE The compiler
- JavaCross 7.0
- Mixed Messages
- Pool Puzzle
- Exercise Solutins: Code Magnets:

- **A Trip to Objectville**

- Chair Wars
- Brain Power
- Making your first object
- Making and testing Movie objects
- Quick! Get out of main!
- Running the Guessing Game
- Who am I?
variable value.
- There are no Dumb Question
- Java Exposed
- Life on the garbage-collectible heap
- Life and death on the heap
- An array is like a tray of cups
- Arrays are objects too
- Make an array of Dogs
- Control your Dog



- A Dog example
- Exercise: BE the compiler
- Exercise: Code Magnets
- Pool Puzzle
- A Heap o' Trouble
- **How Objects Behave**
 - Remember: a class describes what an object knows and what an object does
 - The size affects the bark
 - You can send things to a method
 - You can get things back from a method.
 - You can send more than one thing to a method
 - there are no Dumb Questions
 - Reminder: Java cares about type!
 - Cool things you can do with parameters and return types
 - Encapsulation
 - Encapsulating the GoodDog class
 - How do objects in an array behave?
 - Declaring and initializing instance variables
 - The difference between instance and local variables
 - there are no Dumb Questions
 - Comparing variables (primitives or
- **Using the Java Library**
 - In our last chapter, we left you with the cliff-hanger. A bug.
 - So what happened?
 - How do we fix it?
 - Option one is too clunky
 - Option two is a little better, but still pretty clunky
 - Wake up and smell the library
 - Some things you can do with ArrayList
 - there are no Dumb Questions
 - Java Exposed
 - Comparing ArrayList to a regular
- Five-Minute Mystery
- Exercise Solutions: Code Magnets
- Puzzle Solutions
- references)
- Exercise: BE the compiler
- Who am I?
- Mixed Messages
- Pool Puzzle
- Five Minute Mystery
- Puzzle Solutions
- **Extra-Strength Methods**
 - Let's build a Battleship-style game: "Sink a Dot Com"
 - First, a high-level design
 - The "Simple Dot Com Game" a gentler introduction
 - Developing a Class
 - BRAIN POWER
 - There are no Dumb Questions
 - There are no Dumb Questions
 - Exercise: BE the JVM
 - Exercise: Code Magnets
 - Java Cross
 - Exercise Solutions
- array
- Comparing ArrayList to a regular array
- Let's fix the DotCom code.
- New and improved DotCom class
- Let's build the REAL game: "Sink a Dot Com"
- What needs to change?
- Who does what in the DotComBust game (and when)
- Prep code for the real DotComBust class
- The final version of the Dotcom class



- Super Powerful Boolean Expressions
- Ready-bake Code
- Ready-bake Code
- Using the Library (the Java API)
- You have to know the full name of the class you want to use in your code.
- there are no Dumb Questions
- there are no Dumb Questions
- How to play with the API
- Code Magnets
- JavaCross 7.0
- Exercise Solutions
- JavaCross answers
- **Better Living in Objectville**
 - Chair Wars Revisited...
- **Life and Death of an Object**
 - The Stack and the Heap: where things live
 - Methods are stacked
 - What about local variables that are objects?
 - there are no Dumb Questions
 - If local variables live on the stack, where do instance variables live?
 - The miracle of object creation
 - Construct a Duck
 - Initializing the state of a new Duck
 - there are no Dumb Questions
 - Using the constructor to initialize important Duck state
 - Make it easy to make a Duck
 - Doesn't the compiler always make a no-arg constructor for you? No!
 - there are no Dumb Questions
 - there are no Dumb Questions
 - Wait a minute... we never DID talk about superclasses and inheritance and how that all fits in with

- BRAIN POWER
- there are no Dumb Questions
- there are no Dumb Questions
- brain power
- there are no Dumb Questions
- Exercise: Mixed Messages
- Exercise BE the Compiler
- Exercise Solutions: BE the Compiler
- **Serious Polymorphism**
 - Did we forget about something when we designed this?
 - BRAIN POWER
 - there are no Dumb Questions
 - Pool Puzzle
 - Exercise Solutions
- constructors.
- Making a Hippo means making the Animal and Object parts too...
- How do you invoke a superclass constructor?
- Can the child exist before the parents?
- Superclass constructors with arguments
- Invoking one overloaded constructor from another
- Now we know how an object is born, but how long does an object live?
- What about reference variables?
- **Numbers Matter**
 - MATH methods: as close as you'll ever get to a global method
 - The difference between regular (non-static) and static methods
 - What it means to have a class with static methods.
 - Static methods can't use non-static



- (instance) variables!
 - Static methods can't use non-static methods, either!
 - Static variable: value is the same for ALL instances of the class
 - Initializing a static variable
 - static final variables are constants
 - final isn't just for static variables...
 - there are no Dumb Questions
 - Math methods
 - Wrapping a primitive
 - Before Java 5.0, YOU had to do the work...
 - Autoboxing: blurring the line between primitive and object
 - Autoboxing works almost everywhere
 - But wait! There's more! Wrappers have static utility methods too!
 - And now in reverse... turning a primitive number into a String
 - Number formatting
 - Formatting deconstructed...
 - The percent (%) says, "insert argument here" (and format it using these instructions)
 - The format String uses its own little language syntax
 - The format specifier
 - The only required specifier is for TYPE
 - What happens if I have more than one argument?
 - So much for numbers, what about dates?
 - Working with Dates
 - Moving backward and forward in time
 - Getting an object that extends Calendar
 - Working with Calendar objects
 - Highlights of the Calendar API
- Even more Statics!... static imports
 - Lunar Code Magnets

