First Head Java-First day 4th May 2014

**Chapter 1:**

Java features:

-friendly syntax

-Object-oriented features

- Memory management

-portability

Variables come into two flavors: primitive and reference

**Chapter 2:**

**-**The difference between an object and a class

-what makes OO development in java so fun?

-How objects can improve our life?

Garbage collectible heap! What does it mean?

**Chapter3**:

-what you can declare as a variable?

-what you can put in a variable?

-what you can do with a variable?

**Chapter4:**

**-**How objects behave?

-Objects have **states and behaviors** represented by instance variables and methods how states and behaviors are related.

-object behavior used a unique object state in other words methods use instance variables values

**Chapter5:**

-more methods

-loops and operators

**Chapter6:**

-Pre-built classes in Java

-don’t reinvent the wheel when you know how to use the java library called java API(Application-Programming-Interface)

**Chapter7:**

-Plan your program with the future in mind

-Key features: polymorphism, inheritance

-write the code which can be extended easily the code that will be flexible for those pesky last-minute spec changes

**Chapter8:**

-Going beyond: the inheritance is just the beginning

-to exploit polymorphism we need interfaces

-what is an interface: 100% abstract class

-what is an abstract class: the class that cannot be instantiated!

**Chapter9:**

**-**Life and death of an object (objects are born and object die)

-You are in charge: you decide when and how to construct them. When to abandon them

-garbage collector reclaims the memory

-talking about heap, stack, scope, constructors, super constructors, null references and gc eligibility

**Chapter 10:**

-Do math (formatting numbers and dates)

-methods for absolute values, min, max, rounding

-parsing a string into a number, turning a number into a string

-talk about static variables and methods

**Chapter 11:**

**-**Risky behavior

- handling exceptions

-No matter how good programmer you are stuff can happen; file is not there, server is down when you write a risky method you should take care of bad things that may happen.

**Chapter 12:**

**-**GUI; face it sooner or later you need it!

-for developing some tools

-write client-side code

-see features like Event handling and inner classes

**Chapter 13:**

-swing; the code is easy unless you care actually where everything goes.

-The things that makes it easy to code is the thing that makes it hard to control-The Layout Manager

**Chapter 14:**

-saving objects

-objects can be flattened or inflated.

-objects have state and behavior

-behavior lives in the class but state lives within an individual object.

-how we can save the state; very painful job by interrogating each individual object

-OO way freeze-dry the object (serialize it) and then reconstitute (deserialize it)

**Chapter 15:**

**-**connection with the outside world

- java.net library takes care of every low-level networking details

-the good thing about java is that every sending and receiving data over a network is kind of I/O with different connection stream at the end of the chain.

-how to make client, server, client socket, server socket

**Chapter 16:**

-have all the tools to collecting and manipulating data without having to write sort algorithm

**Chapter 17:**

-organize, package and deploy java code

-deployment options: local, semi-local, remote

**Chapter 18:**

**-**Distributed computing

-Things are easier when all the parts of your application are in one place, in one heap, with one JVM to rule them all.

-How to use RMI (removed method invocation)

-