The Day of celebration	The name of the Show	What will you enjoy learning?
1	**The tale of programming** A walk down the memory lane!	1) Programming - How did it all start and evolve? 2) What should you learn in a programming language? 3) Why is learning a paradigm important? 4) The clash of paradigms 5) Problem solving - the problem space and the solution space
2	**The Objects of Carribean** A film by OO!	1) The object oriented screenplay 2) How do objects interact with each other? 3) Why does complexity occur in software? 4) How does OO manage complexity? 5) What are the features of OOP?
3	**Abstraction** What to show? What not to show?	1) The primary and secondary pillars of OO 2) What is abstraction and why is it needed? 3) What is Interface and Implementation? 4) The abstraction LOC
	Encapsulation Data married to methods!	1) What is encapsulation and why is it needed? 2) Encapsulation in Java 3) What is Information hiding? 4) Abstraction Vs Encapsulation 5) A small case Study
4	**Hierarchy** Pass the wealth!	1) What is hierarchy and why is it needed? 2) Inheritance and Aggregation 3) Benefits of inheritance 4) The drawbacks of inheritance 5) A small case Study

5	**Typing** The mixology!	1) What is typing? 2) Strong typing and weak typing 3) What is polymorphism? 4) Static and dynamic binding
	Concurrency Juggle it together!	1) What is Concurrency? 2) Concurrency is not simultaneity
	Persistence There is life after death!	1) What is persistence? 2) Longevity of objects
6	**Java** The wait is all but over!	1) The Java story 2) The Java World - the platform, JDK, JRE, JVM, JIT 3) The language basics - data types, statements, loops, expressions 4) Type Conversion and Type Casting 5) Shortfilm time
7	**The Object Wars** Coding starts today!	1) How to construct a class to define state and behaviour? 2) Methods and method overloading 3) Objects, Constructors and Constructor overloading 4) Learn how objects are managed in memory 5) Static variables, static methods and static block 6) Using Final to create constants
	Home Assignment-1	It's time to get ya hands dirty: An interesting problem to work on.
8	**Knowledge Session-1**	Review and discuss the assignment. Learn it all in the classroom.

9	**Inheritance and Interfaces - 1** Do not reinvent the wheel!	1) superclass/sublass, supertype/subtype 2) The IS-A and HAS-A test 3) Super, Constructor chaining 4) Early binding(overloading) and Late binding(overriding) 5) Learn how polymorphism is implemented in the real world
10	**Inheritance and Interfaces - 2** Do not reinvent the wheel!	6) Abstract methods and classes 7) Using final with respect to inheritance 8) The Object class 9) Case study on Inheritance 10) Interfaces and their uses
	Home Assignment-2	It's time to get ya hands dirty: An interesting problem to work on.
11	**Knowledge Session-2**	Review and discuss the assignment. Learn it all in the classroom.
12	**Exception Handling** Handle the risk!	1) Understand Exception Handling in a simple way using a scenario 2) Learn the smart way of handling errors - try, catch and finally 3) What is the call stack? 4) Understand Exception class hierarchy 5) Chained exceptions 6) Creating your own exception classes
13 & 14	**Multithreading**	1) Process Vs Thread 2) Life cycle of a Thread 3) Why multithreading? 4) Creating threads 5) Thread synchronization

	Home Assignment-3	It's time to get ya hands dirty: An interesting problem to work on.
15	**Knowledge Session-3**	Review and discuss the assignment. Learn it all in the classroom.
16 & 17	**Innovative features of Java** Adding beauty and power!	1) Packages: containers to the rescue 2) Annotations: the power of metadata 3) Enumerations: adding power to constants 4) Varargs: Variable arguments 5) Autoboxing: automate the conversion from primitive to object wrappers 6) Generics*: catch the bugs early
18 & 19	**Java IO and Serialization**	1) Streams: the abstraction of I/O 2) Byte streams and character streams 3) Buffered I/O: the efficient I/O 4) Scanning and Formatting 5) Standard streams: System.in, System.out and System.err 6) File NIO: The nonblocking IO 7) Serialization: Save the objects
20	**Java Library - Strings and Java.Lang**	1) The String class 2) The StringBuffer class 3) Math class 4) Runtime class 5) Process class 6) ProcessBuilder class 7) System class

21	**Java Utilities** The Collections Framework	1) Collection Interfaces 2) Collection classes - ArrayList, Vector, Hashset, Treeset 3) Comparator Interface 4) Collections algorithms
22	**The Design Theory**	1) How to solve bigger problems using OOP? 2) How to learn the art of identify classes? 3) How to know what methods to add? 4) How to build true OO solutions? 5) Case Study
23	**Class Assignment** followed by **Home Assignment-4**	Design the interface in the class. Build the implementation at home.
24	**The Coding Hour-1**	In-class demonstration of solving an interesting problem. 1) Learn how to approach a problem 2) Learn how to go about selecting the required tools to solve the problem. 3) Learn how to go from design to implementation.
25	**Network Programming**	1) Basics of network programming 2) Client and server programming
26	**Applets** Let's not disturb the dead	Let's accept the fact - Applets are no longer used as much. Flash, Javascript, Actionscript deliver a lot more than applets - An applet program (Just for the sake of it)

27 & 28	**Java UI** Lets swing!	1) Using top level containers 2) Components and Layout Managers 3) Event Handling 4) Using swing components 5) Using Borders
29	**The Coding Hour-2**	In-class demonstration of solving an interesting problem. 1) Learn how to approach a problem 2) Learn how to go about selecting the required tools to solve the problem. 3) Learn how to go from design to implementation.
	Home Assignment-5	It's time to get ya hands dirty: An interesting problem to work on.
30	**Knowledge Session-4**	Review and discuss the assignment. Learn it all in the classroom.
31	**The Coding Hour-3**	In-class demonstration of solving an interesting problem. 1) Learn how to approach a problem 2) Learn how to go about selecting the required tools to solve the problem. 3) Learn how to go from design to implementation.
32	**Knowledge Session-5**	Review and discuss the assignment. Learn it all in the classroom.
33	**New features of Java in 8.0**	Looking at all the new features of Java that were added in 8.0
34	**Wrapping it up!** Time to bid farewell!	1) How to continue the journey of learning and get to the next level? 2) How to master the Java library? 3) How to solve substancially bigger problems? □