



# Object-Oriented Programming (CS F213)

## Module I: Object-Oriented and Java Basics

**BITS Pilani**

**Dr. Pankaj Vyas**

Department of Computer Science, BITS-Pilani, Pilani Campus

# Today's Agenda

---



- Course Overview
- What is an Object ?
- Graphical View of an Object
- Object Examples
- What is a Class ?
- Object vs Class ?
- Class Examples

# Course Overview



- **Text Books**

1. **Java: The Complete Reference, Herbert Schildt, McGraw Hill Education and Oracle Press, Ninth Edition, 2014.**

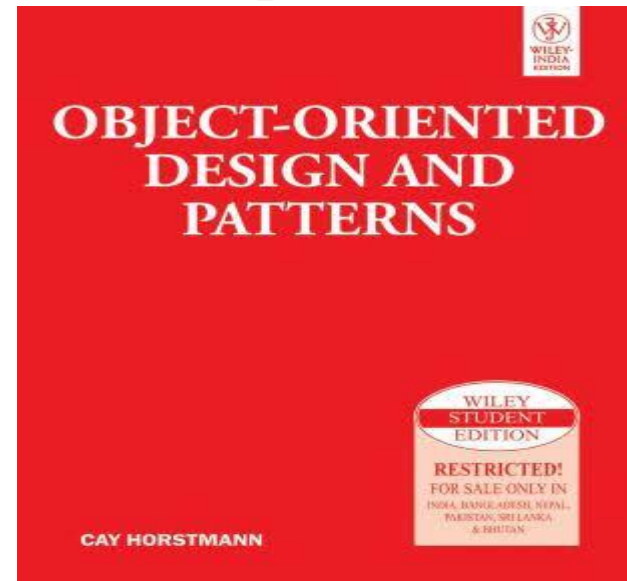


# Course Overview



- **Text Books**

1. **Java: The Complete Reference, Herbert Schildt, McGraw Hill Education and Oracle Press, Ninth Edition, 2014.**
2. **Object Oriented Design & Patterns, Cay Horstmann, John Wiley & Sons, 2004 [Must Procure]**



# Course Overview ..



## Evaluation Components

# Course Overview ..



## Evaluation Components

Component	Duration	Date & Time	%Weight	Nature
Mid Semester Test	90 Minutes	Saturday , 11/03/2017 Time: 9:00 AM – 10:30 PM [Syllabus: Lecture 1 – 21]	25% [75 Marks]	CB

# Course Overview ..



## Evaluation Components

Component	Duration	Date & Time	%Weight	Nature
Mid Semester Test	90 Minutes	Saturday , 11/03/2017 Time: 9:00 AM – 10:30 PM [Syllabus: Lecture 1 – 21]	25% [75 Marks]	CB
Lab Attendance	There will be 12 Labs. Each Lab Carries 1.5 Marks Marks for 10 Labs Will be Considered in the End		5% [15 Marks]	-

# Course Overview ..



## Evaluation Components

Component	Duration	Date & Time	%Weight	Nature
Mid Semester Test	90 Minutes	Saturday , 11/03/2017 Time: 9:00 AM – 10:30 PM [Syllabus: Lecture 1 – 21]	25% [75 Marks]	CB
Lab Attendance	There will be 12 Labs. Each Lab Carries 1.5 Marks Marks for 10 Labs Will be Considered in the End		5% [15 Marks]	-
Online Test	120 Minutes	16 <sup>th</sup> April, 2017 (Sunday) Time: 3:00 PM – 6:00 PM [Syllabus: Multithreading + GUI + Event Handling ]	30% [90 Marks]	OB



# Course Overview ..



## Evaluation Components

Component	Duration	Date & Time	%Weight	Nature
Mid Semester Test	90 Minutes	Saturday , 11/03/2017 Time: 9:00 AM – 10:30 PM [Syllabus: Lecture 1 – 21]	25% [75 Marks]	CB
Lab Attendance	There will be 12 Labs. Each Lab Carries 1.5 Marks Marks for 10 Labs Will be Considered in the End		5% [15 Marks]	-
Online Test	120 Minutes	16 <sup>th</sup> April, 2017 (Sunday) Time: 3:00 PM – 6:00 PM [Syllabus: Multithreading + GUI + Event Handling ]	30% [90 Marks]	OB
Comprehensive	180 Minutes	Monday, 15 <sup>th</sup> May, 2017	40% [120 Marks]	OB

# Course Overview ..



## Object-Oriented Lab Sessions

Lab #	Topics to be Covered
1	Moving from C to Java [Java Basics]
2	Class Design Basics I
3	Class Design Basics II
4	Packages, Inheritance and Polymorphism
5	Arrays and Strings
6	Interfaces, Inner classes and anonymous inner classes
7	Exception Handling and Collections
8	JAVA GUI and Event Handling
9	File Handling
10	Multi-Threading-1
11	Class Design Lab
12	Design Pattern Lab

- There will be 12 Lab Sessions.
- Each Lab Carries 1.5 Marks. [Total : 15 Marks (10 out of 12)]

# Course Overview .. Handout



## Module I : Object-Oriented and Java Basics

### Lecture 1: Object-Oriented Basics

- Object and Class Basics
- Basic Pillars of Object-Oriented Programming (Abstraction, Encapsulation, Inheritance and Polymorphism]

### Lecture 2-3: Java Programming Syntax

- Java Program Structure, Compiling and Executing a Simple Java Application
- Types of Variables in Java
- Primitive Types in Java
- Type Promotion and Type Casting

### Lecture 4-5 : Defining Classes and Object Creation

- Defining Classes and Access Modifiers, Creating Objects, Role of Constructors
- Accessing Instance Fields and Methods
- Local Variables vs Instance Fields , Mutable and Immutable Objects
- Command-Line Arguments, Reading Input from console Using Scanner class

### Lecture 6: Use of static final keywords in Java Method Overloading

- Use of static and final keywords in Java
- Method Overloading

### Lecture 7: Objects as Parameters

- Objects as Parameters to Methods and Object class in Java

# Course Overview .. Handout



## Module II : Arrays and String in Java

### Lecture 8-9 : Arrays in Java

- Implementing 1-D and 2-D Arrays in Java, Role of Arrays class
- Implementing Dynamic Arrays Using Vector class

### Lecture 10-11: Strings in Java

- String class, Important String Methods
- StringBuffer and StringTokenizer class in Java

## Module III : Polymorphism and Inheritance in Java

### Lecture 12-13 : Inheritance in Java

- Extending classes and Role of super keyword
- Method Overriding [Super Type vs Sub-Type Relationships]

### Lecture 14-15: Abstract Classes, Abstract Methods and Interfaces

- Abstract Methods and Classes
- Interfaces in Java [ class vs interface]
- Comparable and Comparator Interfaces in Java
- Nested and Inner Classes

### Lecture 16: Generic Programming

- Generic Form of a Class
- Generic Interfaces and Bounded Types

## Module IV: Exception Handling Mechanism

### Lecture 17-18 : Exceptions in Java

- Exception Basics and Types
- Catching Exceptions
- Writing Your Own Exceptions

# Course Overview .. Handout



## Module V: Collections Framework of Java

### Lecture 19-21 : Collections in Java

- Introduction to Collection Framework in Java, Important Collection Interfaces and Their Methods
- ArrayList and LinkedList Classes in Java
- Iterators and ListIterators
- Wrapper classes and Autoboxing

## Module VI: Multithreaded Programming in Java

### Lecture 22-24: Multithreading

- Multithreading vs Multitasking
- Thread Class in Java and its Important Methods
- Creating Your own Threads and Runnable Interface
- Thread Synchronization, Inter Thread Communication
- Suspending and Resuming Threads

## Module VII: GUI Programming

### Lecture 25-27: GUI Programming with Swing

- Introduction to swing package
- Containers and Components and Layouts and LayoutManager Interface
- JLabel class, JTextField class
- Swing Buttons, JButton, JToggleButton
- Check Boxes, Radio Buttons
- JScrollPane , JMenu, JMenuBar and JMenuItem
- Designing Frames and Adding Components, Timer Class in Java



## Module VIII: Event Handling in Java

### Lecture 28-30: Event Handling

- Delegation Event Model
- Event Classes, Listener Interfaces
  - ActionEvent and AdjustmentEvent Classes
  - ComponentEvent and ContainerEvent Classes
  - FocusEvent and InputEvent Classes
  - MouseEvent and ItemEvent Classes
- Listener Interfaces
  - ActionListener and AdjustmentListener Interfaces
  - ComponentListener and ContainerListener Interfaces
  - FocusListener and ItemListener Interfaces
  - MouseListener and MouseMotionListener

# Course Overview .. Handout



## Module IX: Object-Oriented Analysis and Design

### Lecture 31-33 : Object-Oriented Analysis

- Object Relationships and their representation in UML
- What are Use-Case Models and Use-Case Realization Templates
- UML Activity Charts
- Identifying Classes Using Noun-Phrase Analysis

### Lecture 34-35: Object-Oriented Design

- Goals of Object-Oriented Design Phase
  - Identifying Attributes and Methods of Each class
  - Class Diagram, Sequence Diagrams, State Diagrams

## Module X: Object-Oriented Patterns

### Lecture 36-40: Object-Oriented Design Patterns

- Design Pattern Basics
  - Creational Patterns (Singleton Pattern , Factory Pattern , Factory Method Pattern)
  - Structural Patterns (Adapter Pattern, Composite Pattern, Decorator Patterns, Proxy Pattern)
  - Behavioral Patterns (Iterator Pattern, Chain of Responsibility, Strategy Pattern, Proxy Pattern, Visitor Pattern, Command Pattern )

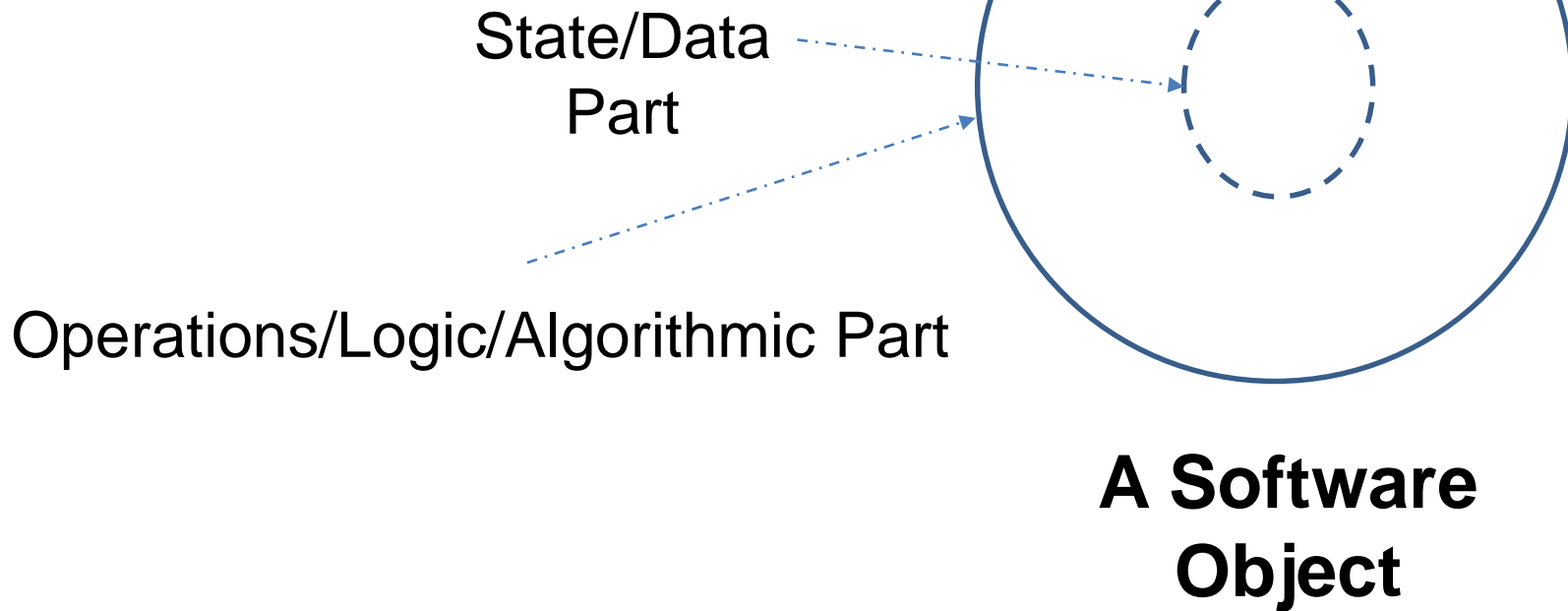
# What is Object ?

- Object Means Combination of Data **(Attributes)** and Logic **(Algorithm, Behavior, Functions, Operations)** of some real world entity. For example Student, Box, Account, Time
- Every real-world object has two characteristics :
  - Data-Part/State [Also known as attributes or properties]
  - Behavior [Also known as operations / Algorithmic / Logic Part]
- Software Object is conceptually similar to a every real-world object.



# Object: Graphical View

**Object Keeps State/Data Part and Behavior/Logic Part Together**



# Object Examples

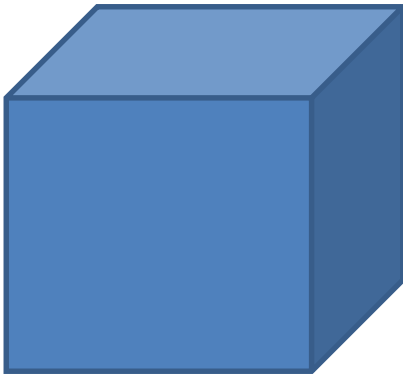


- **Box Object :**
  - State/Data Part : length, width, height, Color [Attributes/Instance Fields]
  - Behavior Part: computing area, computing volume [Operations, Methods]
- **Dog Object :**
  - State/Data Part : name, breed, color [Attributes / Instance Fields]
  - Behavior Part : barking, fetching, wagging [Operations, Methods]
- **Account Object :**
  - State/Data Part : account number, account holder name, balance, type of account [Attributes/ Instance Fields]
  - withdrawing an amount, depositing an amount, checking balance of a account [ Behavior, Operations, Methods]

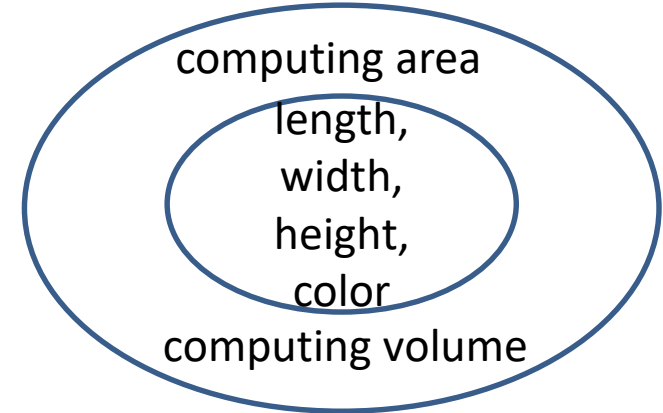
# Object Examples



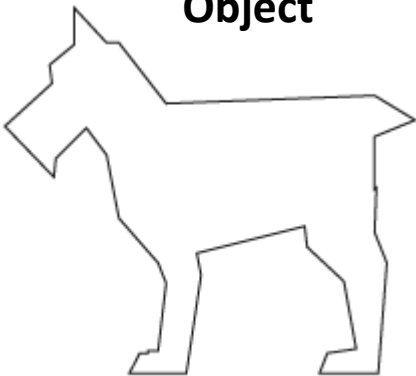
**Box as a Real World  
Object**



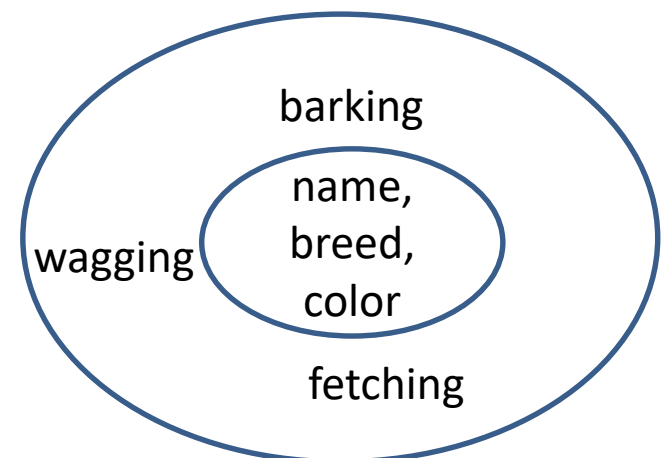
**Box as a Software Object**



**Dog as a Real World  
Object**



**Dog as a Software Object**



# What is Class ?

---

- Objects are grouped in classes
- A class is a collections of objects having similar behavior and attributes
- An object is simply a single instance of class.
- Objects can not be instantiated (or created) without defining a class
- Classes are defined whereas objects are created.
- In order to create an object, you have to first define the class of that object

# Class Example : Box Class

```
class Box
```

```
{
```

```
    private    double    length;
```

```
    private    double    width;
```

```
    private    double    height;
```

Instance Fields

```
    public     double    area()
```

```
    {
```

```
        return 2* (length * width + width * height + height * length);
```

```
    }
```

```
    public     double    volume()
```

```
    {
```

```
        return length * width * height;
```

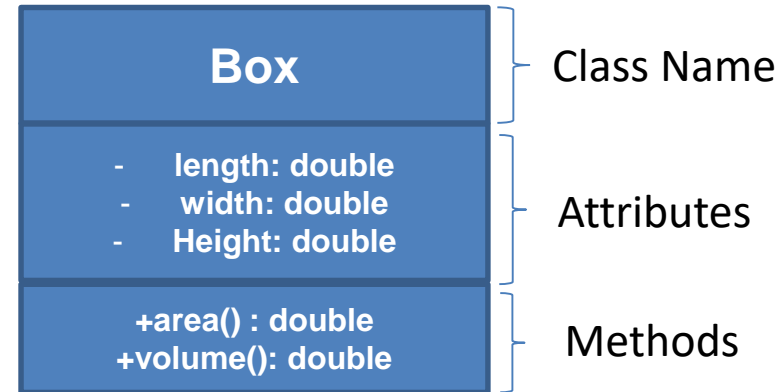
```
    }
```

Methods

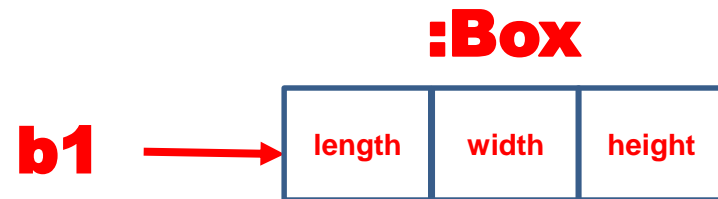
```
// End of class
```

# Class Example : Box Class

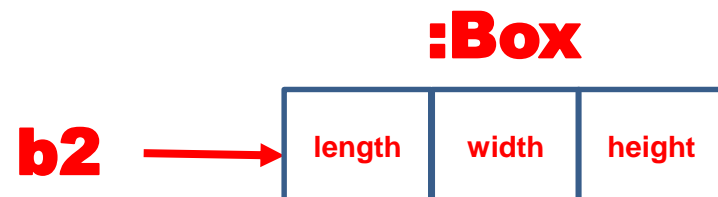
```
class Box
{
    private double    length;
    private double    width;
    private double    height;
    public double     area()
    {
        return 2* (length * width + width * height + height * length);
    }
    public double     volume()
    {
        return length * width * height;
    }
}
// End of class
```



**Box b1 = new Box();**



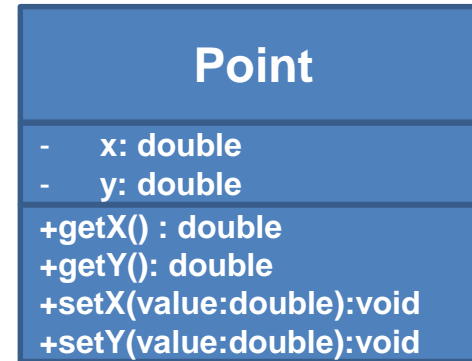
**Box b2 = new Box();**



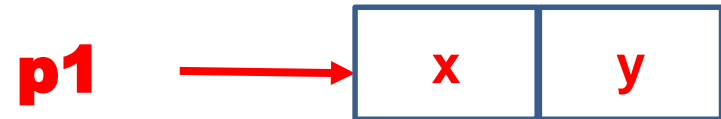
# Class Example : Point



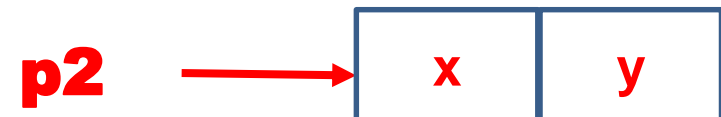
```
class Point
{
    private double    x;
    private double    y;
    public double      getX()
    {
        return        x;
    }
    public double      getY()
    {
        return        y;
    }
    public void         setX(double value)
    {
        x = value;
    }
    public void         setY(double value)
    {
        y = value;
    }
} // End of class
```



**Point p1 = new Point();**  
**:Point**



**Point p2 = new Point();**  
**:Point**



# Exercise



- Think about the class named 'Student'. Define its state and operations
- Define the instance fields (attributes) and methods of class named 'Line'.