



Monsoon 2019

Lecture - 1

Object Oriented Programming

by

Dr. Rajendra Prasath

Indian Institute of Information Technology

Sri City - 517 646, Andhra Pradesh, India

Introduction



- ✧ Can you visualize everything in terms of Objects?
 - ✧ Yes / No
 - ✧ Why so?
- ✧ Is it using Objects?
- ✧ using C++, Java, C#, Smalltalk?
- ✧ What makes a program Object Oriented?
- ✧ How do you measure a good design?

What is OOP?



✧ Object Oriented Programming:

- ✧ Programming will be the main focus in such a way where everything is an object that interact with one another
- ✧ Designing Classes & Objects
 - ✧ An incremental, and iterative processes
 - ✧ Is it difficult to design right from the first time
- ✧ Is it easy to make changes?
- ✧ Flexibilities in adapting Object Oriented Design and Analysis
- ✧ Many more facets of OOPs

What is an Object?



_____↑

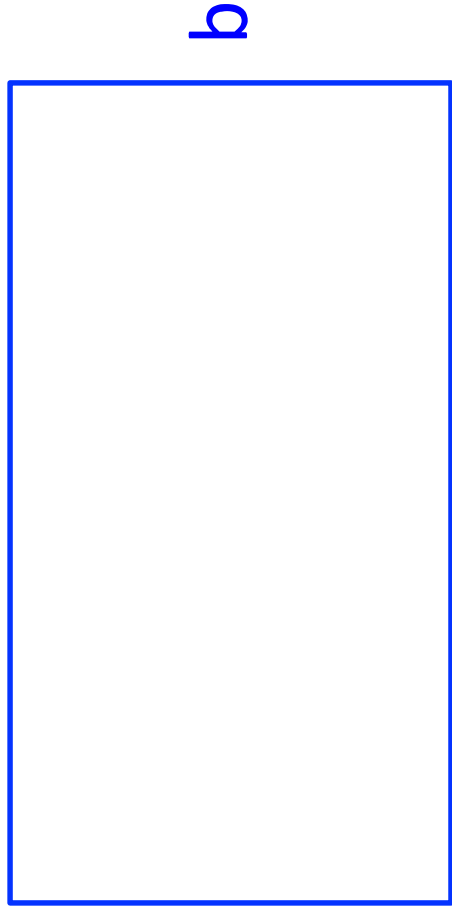
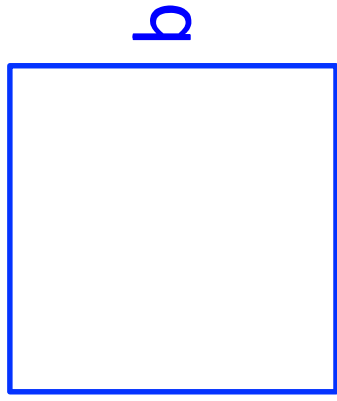
_____↓

_____↑

[_____]

[_____)

Objects?

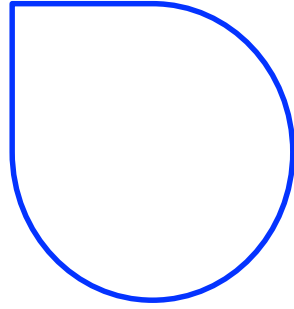
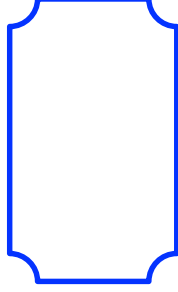
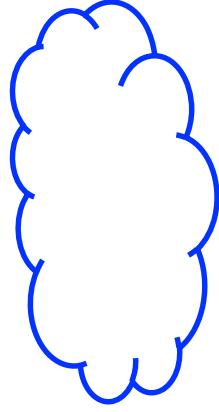
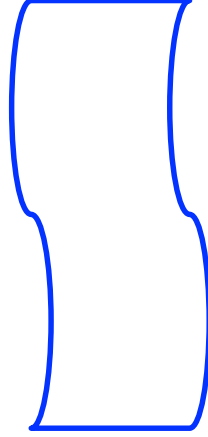
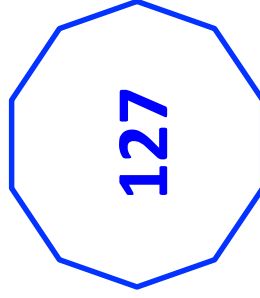


a

b

a

b



31/07/19

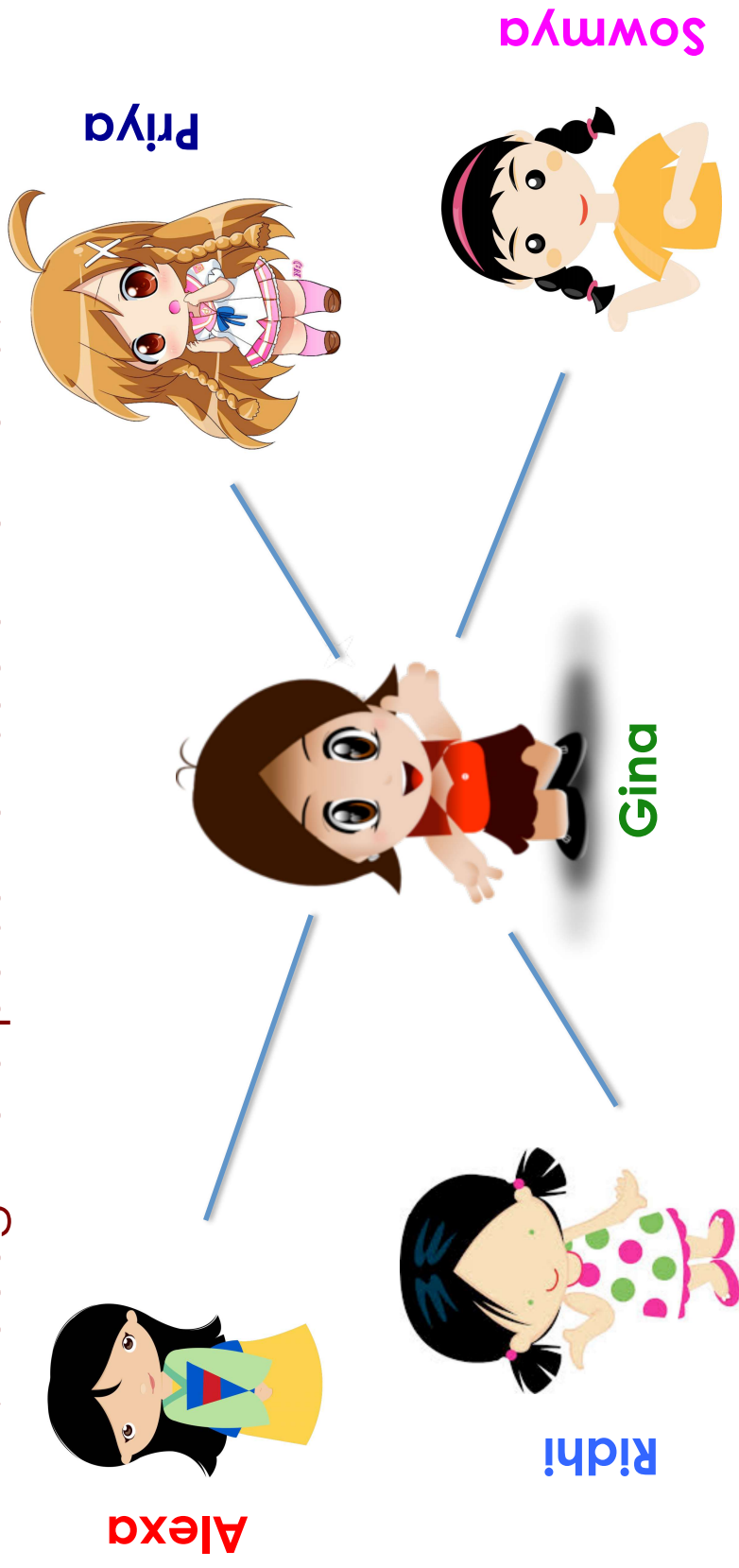
5

Objects – Relations?



Graphs - Networks

- Different Networks
- Ex: Recognize a person on social networks?



- Friend of a Friend is also a Friend (Mutual Friend)



OOP Concepts

- ✧ Introduction
- ✧ Basics of Computing
- ✧ Arrays and Pointers
- ✧ Programming Languages, Compiling
- ✧ Object Oriented Design
- ✧ Classes, Objects, Inheritance
- ✧ Encapsulation
- ✧ Abstract Classes



OOP Concepts

- ✧ Polymorphism
- ✧ Planning, Design
- ✧ Debugging
- ✧ Event Driven Computing, Graphics
- ✧ Sorting, Searching
- ✧ Theory of Computation
- ✧ Algorithmic and Problem Solving Strategies
- ✧ Garbage Collection Strategies



Two Steps to Remember

- ✧ **Data Structures**
 - ✧ The choice of Data Structures
 - ✧ Built-in Data Structures (Primitive)
 - ✧ User Defined Data Structures (Abstract)
- ✧ **Computational Efficiency**
 - ✧ Time Complexity
 - ✧ Space Complexity
 - ✧ Problem / Solution Specific Constraints
 - ✧ Best Practices / Efficient Approaches

Course Content

- Course is divided into several modules:

Module: M1 – M3 and M4

- Covers Basic OOP to Advanced OOP (at least one example problem with detailed analysis)
- Course is supposed to be an interactive course and class performance bonus would be given to students who solve the given set of problems efficiently

➔ Course Content follows ...





M1: Fundamentals

- ✧ Introduction
- ✧ Basics of OOP, Basic Java programs, programming environment, program control
- ✧ Basic Computation - Flow of Control: Branching
- ✧ Program Control, Classes and Methods
- ✧ Flow of Control Loops
- ✧ Defining Classes and Methods, Packages, access specifiers, composition



M2: Intermediate

- ✧ Constructors, finals, class loading
- ✧ accessor/mutator (getter/setter) methods
- ✧ Objects and methods
- ✧ Inheritance
- ✧ Abstract classes
- ✧ Inheritance
- ✧ Polymorphism
- ✧ Interfaces and inner classes
- ✧ Benefits of Object Oriented Programming Methodologies



M3: Needed Components

- ✧ Exception Handling, Streams
- ✧ File I/O, and Networking
- ✧ Arrays
- ✧ Collections and Iterators
- ✧ Dynamic Data Structures and Generics
- ✧ Recursion
- ✧ Threads
- ✧ JDBC with MySQL or NoSQL



M4: Applications Development

- ✧ Case Studies:
 - ✧ Open Source Search Engines
 - ✧ NoSQL frameworks
- ✧ Big Data frameworks - underlying code and code walkthrough
- ✧ and industrial standards
- ✧ Developing Scalable Applications using OOP
 - ✧ Do you think of any project ?
- and many more ...

Learning Outcome - 1



- ✧ Explain the steps in creating an executable program
- ✧ Focus on the intermediate representations and their purpose
- ✧ Apply good programming style and understand the impact of style on developing and maintaining programs
- ✧ Effectively use a version control system and the Linux command line tools for incremental development

Learning Outcome - 2



- ✧ Explain the benefits of object oriented design and understand when it is an appropriate methodology
- ✧ Design object oriented solutions for small systems involving multiple objects
- ✧ Implement, test and debug solutions in JAVA
- ✧ Identify the relative merits of different algorithmic designs
- ✧ Independently find and interpret discipline related documentation
- ✧ Explain the relevance of ethics in the context of Software Engineering

Textbooks



- ✧ Paul Deitel and Harvey Deitel. 2011. **Java how to Program (9th ed.)**. Prentice Hall Press, Upper Saddle River, NJ, USA
- ✧ David J. Eck. 2009. Programming: Introduction to Programming Using JAVA. CreateSpace, Paramount, CA
- ✧ Szyperski, C., Gruntz, D., Murer, S. (2002). Component Software: Beyond Object-Oriented Programming. ACM Press and Addison-Wesley.
- ✧ Any decent material that clearly illustrates OOP Concepts
- ✧ **State-of-the-art approaches:**
 - ✧ **Research Papers / Seminar Papers / Case Studies**



Lab Based Assignments

- ✧ Solve a set of problems every week
- ✧ Must be solved by individuals
- ✧ Must be finished before the deadline specified for that set of problems
- ✧ All Assignments are COMPULSARY
- ✧ Total Weightage: 40%

✧ **NOTE:**

- ✧ if you fail to explain your solution, you will get “0”
- ✧ Solutions would be cross checked !!
- ✧ Solutions submitted after the deadline will not be considered for evaluation
- ✧ Submission Procedure would be given

Grading Pattern

- ✧ Relative Grading
- ✧ Deserving Students – Deserving Grades

✧ Examinations

- ✧ Mid Semester – 1: 10 Marks
- ✧ Mid Semester – 2: 10 Marks
- ✧ End Semester : 20 Marks

- ✧ Total Weightage (100) = Take Home Assignments (40) + Exams (40) + Best Solutions (5) + Specific Task Completion (15)

- ✧ Academic Code of Conduct
- ✧ Explore PENALTIES



Good / Bad Practices



- ✧ Some times hard to realize this
- ✧ It is much easier to copy, paste and modify code to get it working the way you want it, isn't it
- ✧ Duplicating code results in
 - ✧ Poor maintainability
 - ✧ Expensive to fix bugs/errors
 - ✧ Hard to keep up with change
- ✧ Anything you try in your own is a GOOD practice !!



Penalties

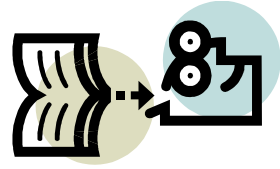
- ✧ Every Student is expected to strictly follow a fair Academic Code of Conduct to avoid severe penalties
- ✧ Penalties would be heavy for those who involve in:
 - ✧ **Copy and Pasting** the code
 - ✧ **Plagiarism** (copied from your neighbor or friend – in this case, both will get “0” marks for that specific take home assignments)
 - ✧ If the candidate is **unable to explain his own solution**, it would be considered as a “copied case” !!
 - ✧ **Any other unfair means** of completing the assignments

Assistance

- ✧ You may post your questions to me at any time
- ✧ You may meet me in person on available time or with an appointment
- ✧ You may leave me an email any time
(email is the best way to reach me faster)



Thanks ...



... Questions ???

