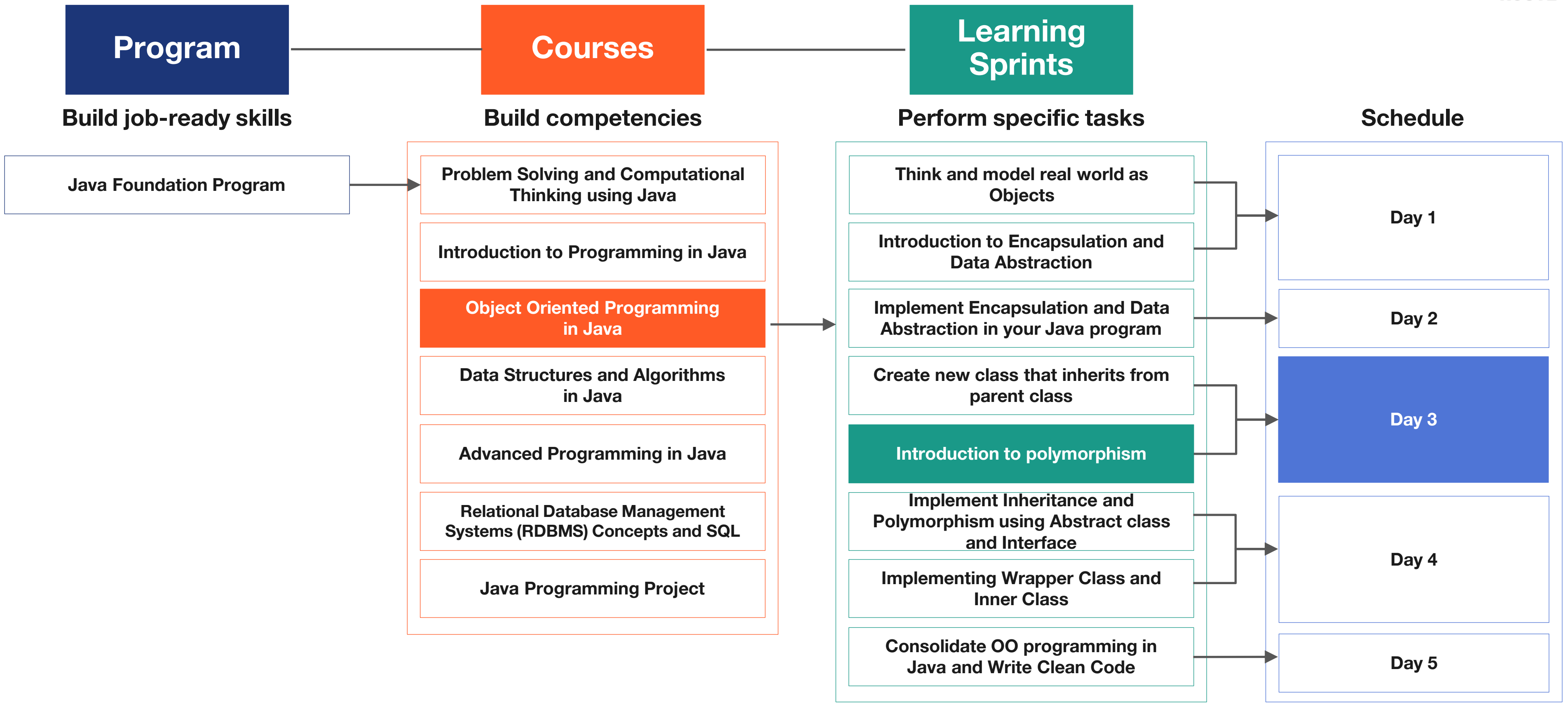
Java Program: Course 3: Plan



Program
Build job-ready skills

- Java Foundation Program

Courses
Build competencies

- Problem Solving and Computational Thinking using Java
- Introduction to Programming in Java
- **Object Oriented Programming in Java**
- Data Structures and Algorithms in Java
- Advanced Programming in Java
- Relational Database Management Systems (RDBMS) Concepts and SQL
- Java Programming Project

Learning Sprints
Perform specific tasks

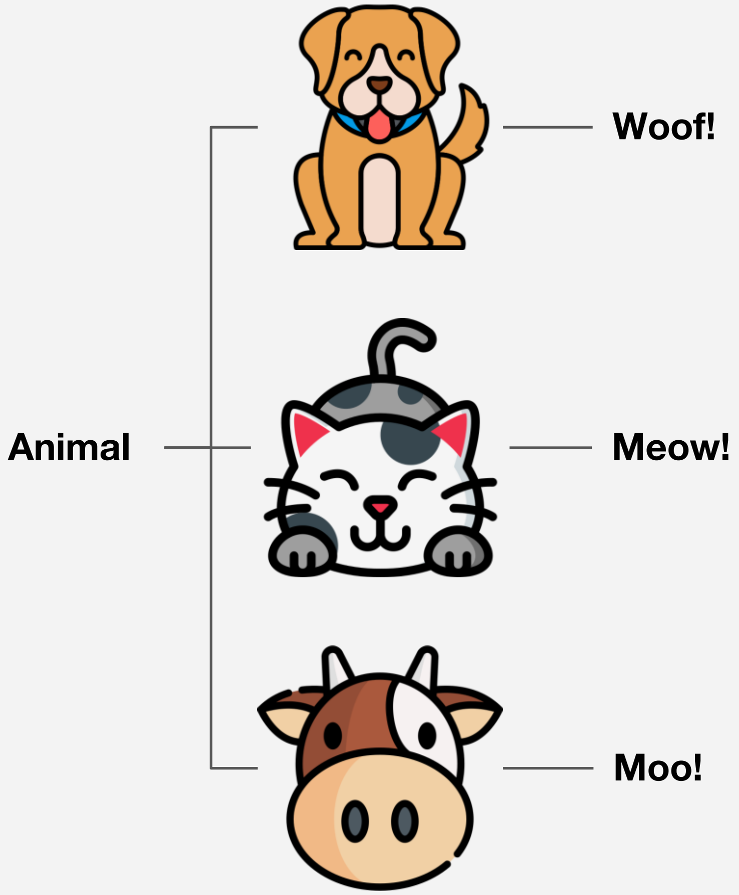
- Think and model real world as Objects
- Introduction to Encapsulation and Data Abstraction
- Implement Encapsulation and Data Abstraction in your Java program
- Create new class that inherits from parent class
- **Introduction to polymorphism**
- Implement Inheritance and Polymorphism using Abstract class and Interface
- Implementing Wrapper Class and Inner Class
- Consolidate OO programming in Java and Write Clean Code

Schedule

- Day 1
- Day 2
- **Day 3**
- Day 4
- Day 5

Think and Tell

Can you list the common behaviors of these animals?

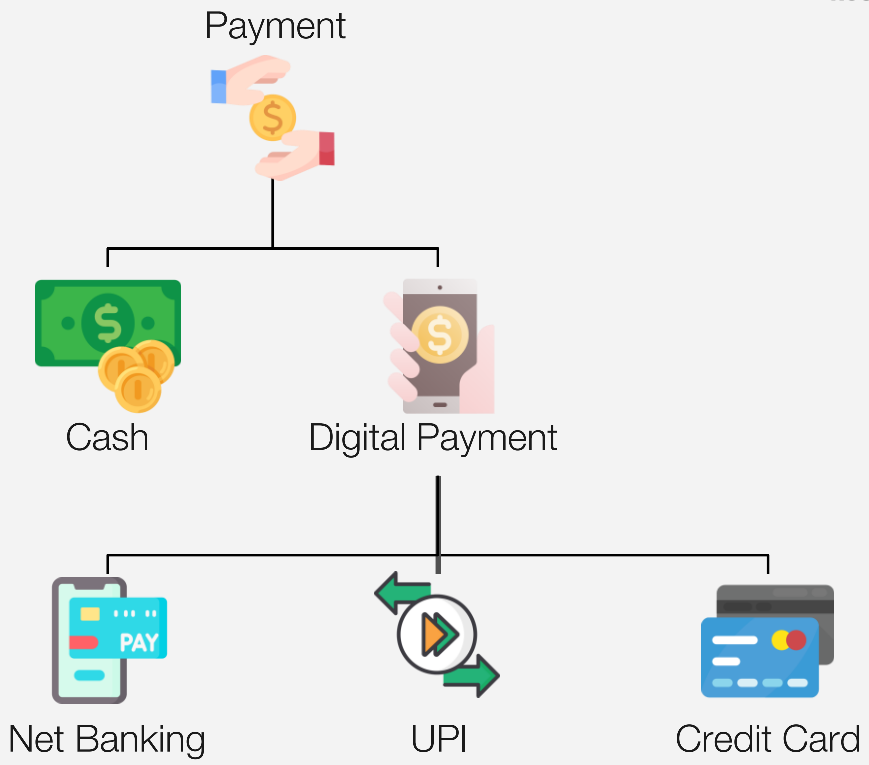


Human Behavior

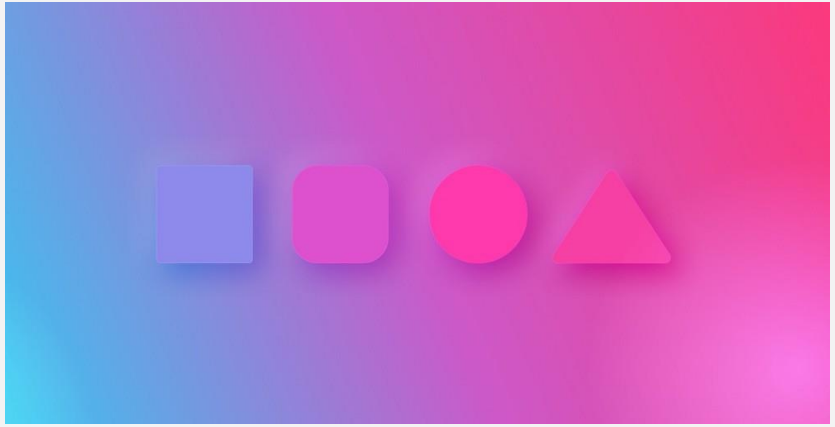
How does a human behavior change in different environments and in different roles?

Application Behavior

What is the common behavior of these applications?



Shapes



What is common among these shapes?

How does an object respond differently to different messages or actions?

How can we implement a single functionality of objects differently in different situations?

Introduction to Polymorphism

Learning Objectives

- Define polymorphism
- Identify the types of polymorphism
- Differentiate between static and dynamic polymorphism
- Implement polymorphism

Introduction to Polymorphism

- Polymorphism is a combination of two words, “poly” means “many” and “morph” means “forms”
- It is an object-oriented programming feature that enables an entity to exist in multiple forms
- It is the ability of an object to make more than one form

Types of Polymorphism

- In Java, polymorphism is of two types:

Static polymorphism

Dynamic polymorphism

Static Polymorphism

- In static polymorphism:
  - An entity, such as a method, can exist in multiple forms
  - One or more methods can exist with the same name but with a different argument list
- When static polymorphism is exhibited by methods, it is known as method overloading

Static Polymorphism (contd.)

- The following code snippet overloads the “calculate” method

```
calculate( int x, int y)
{
        /* Some code to be added */
}
calculate (float x, int y, int z)
{
        /* Some code to be added */
 }
```

Static Polymorphism (contd.)

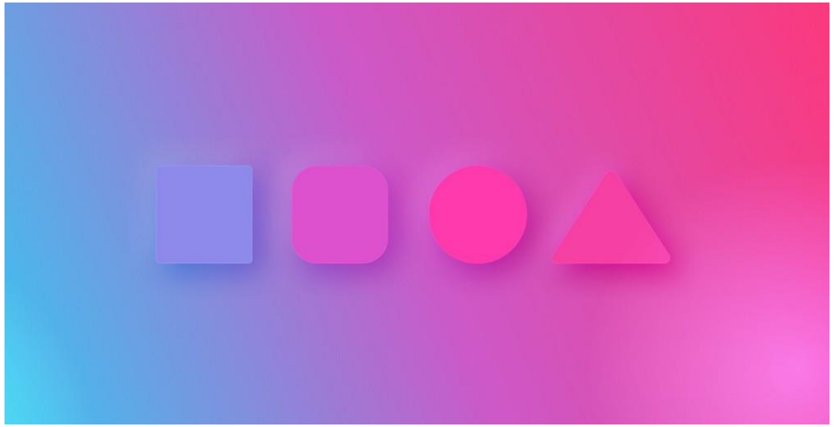
While implementing method overloading, it is important to consider the following points about overloaded methods.

Overloaded methods differ in

- The type and/or the number of their arguments
- The sequence of their parameters
- The data types of their parameters

Interactive Demo

Create a program to calculate the area of the following shapes by implementing method overloading.



Dynamic Polymorphism

- In Java, if a superclass and a subclass contain methods with the same name, the version to be invoked is decided by the JVM at runtime
- The preceding decision to invoke the appropriate method is known as dynamic polymorphism
- Dynamic polymorphism is implemented in Java by method overriding
- Method overriding enables a subclass to provide its own implementation of a method that already has an implementation defined in its superclass
- To override a method present in the superclass, the subclass method should have the same name, same parameters, and same return type as the method in the superclass

Dynamic Polymorphism contd.

The “showDetails” method can be overridden by using the following code snippet:

```
class Person
{
        public void showDetails()
        {
        }
}
class Employee extends Person
{
        public void showDetails()
        {
        }
}
```

Dynamic Polymorphism (contd.)

- Points to consider while implementing overriding:
  - Private methods cannot be overridden as they are not accessible in subclasses
  - Final methods cannot be overridden
  - An overridden method cannot be granted more restrictive access rights in a subclass than it is assigned in case of a superclass

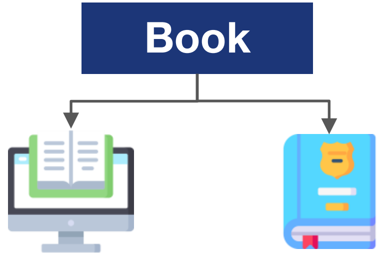
Interactive Demo

Write a program to create a library management system that will store the details of the e-books and paper books.

Task 1: For e-books, accept the author name, title, number of pages, price, book format, and the application needed to open the e-book.

Task 2: For paper books, accept the author name, publisher name, title, number of pages, and price.

Task 3: Display the details of the e-books and paper books by implementing code reusability.



Key Takeaways

- Polymorphism and its usage
- Static and dynamic polymorphism
- Implementing polymorphism

Thank you!