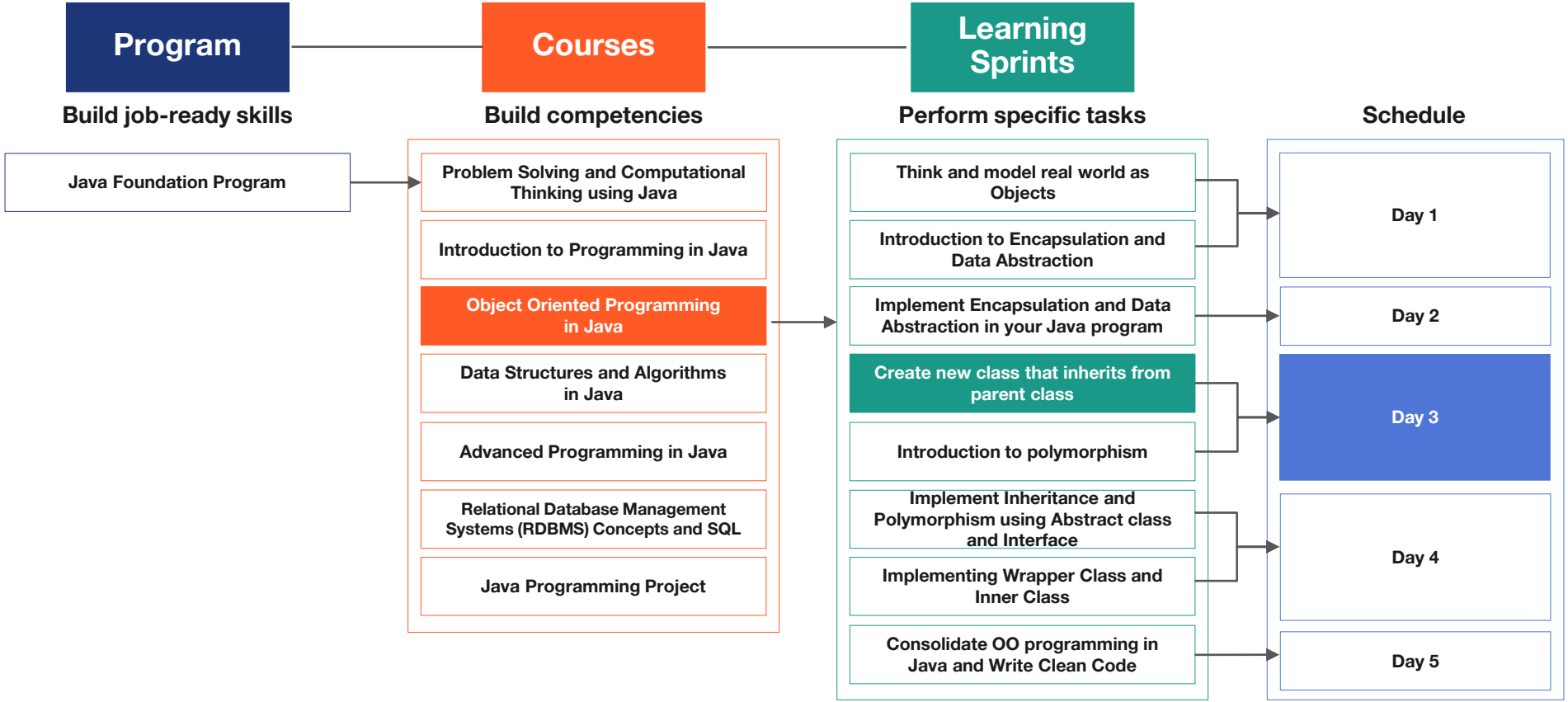
Java Program: Course 3: Plan



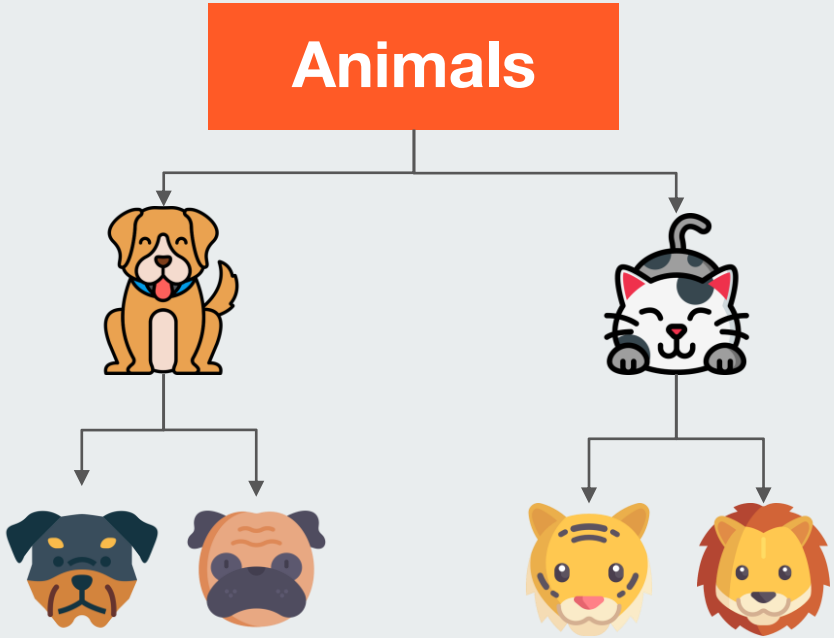
Animal Species

Observe the animal classification.

How do you think the different species of the dog family are related?

What features do they share?

Which feature is unique in each one of them?

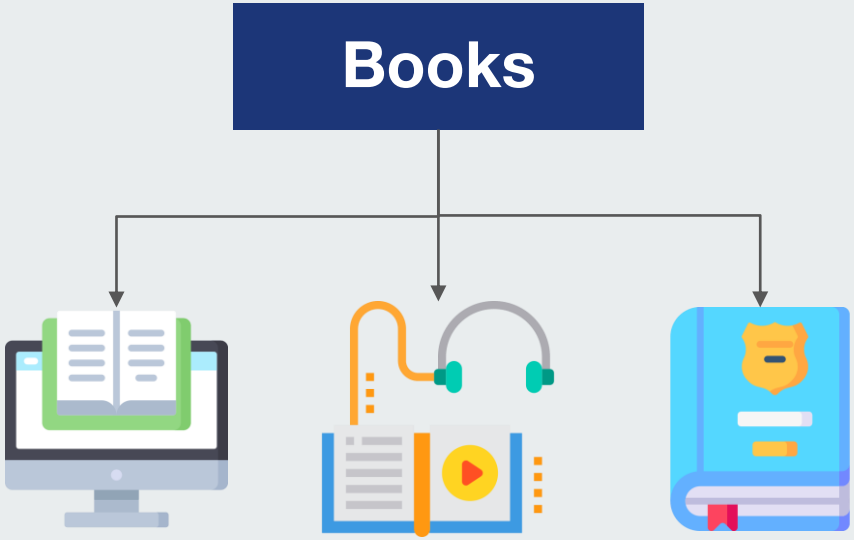


Book Formats

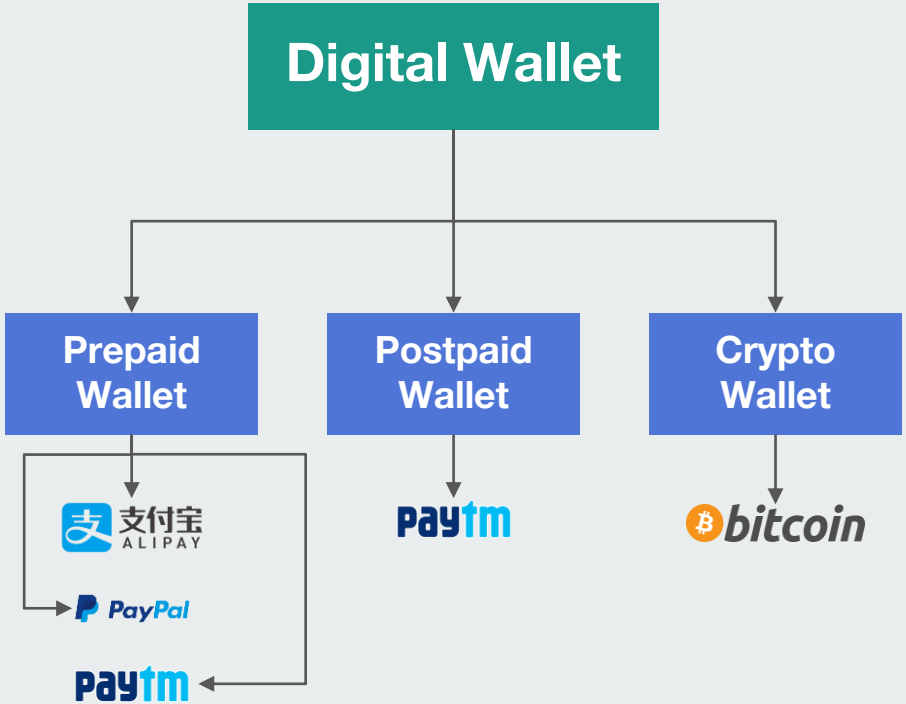
Books come in various formats.

What common characteristics do they have?

How are they different?



Digital Wallet



How are digital wallets similar and different at the same time?

Relationship Between Classes

How can we implement the classifications we spoke about in the above examples in a program?

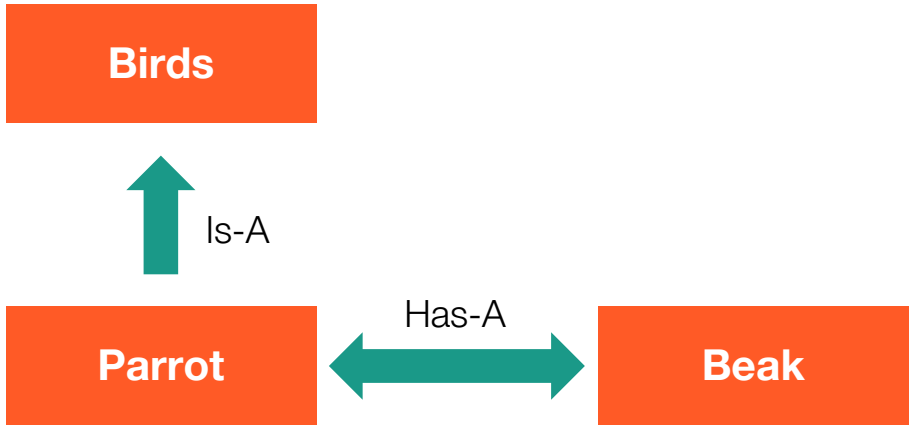
Create New Class That Inherits from Parent Class

Learning Objectives

- Define inheritance
- Differentiate between the types of inheritance: Single level, Multi level, Hierarchical, and Hybrid
- Implement inheritance

Types of Relationships Between Classes

- Has-A (Composition and Aggregation): Implemented by using objects
- Is-A (Inheritance): Implemented by using the “extends keyword” in Java



Introduction to Inheritance

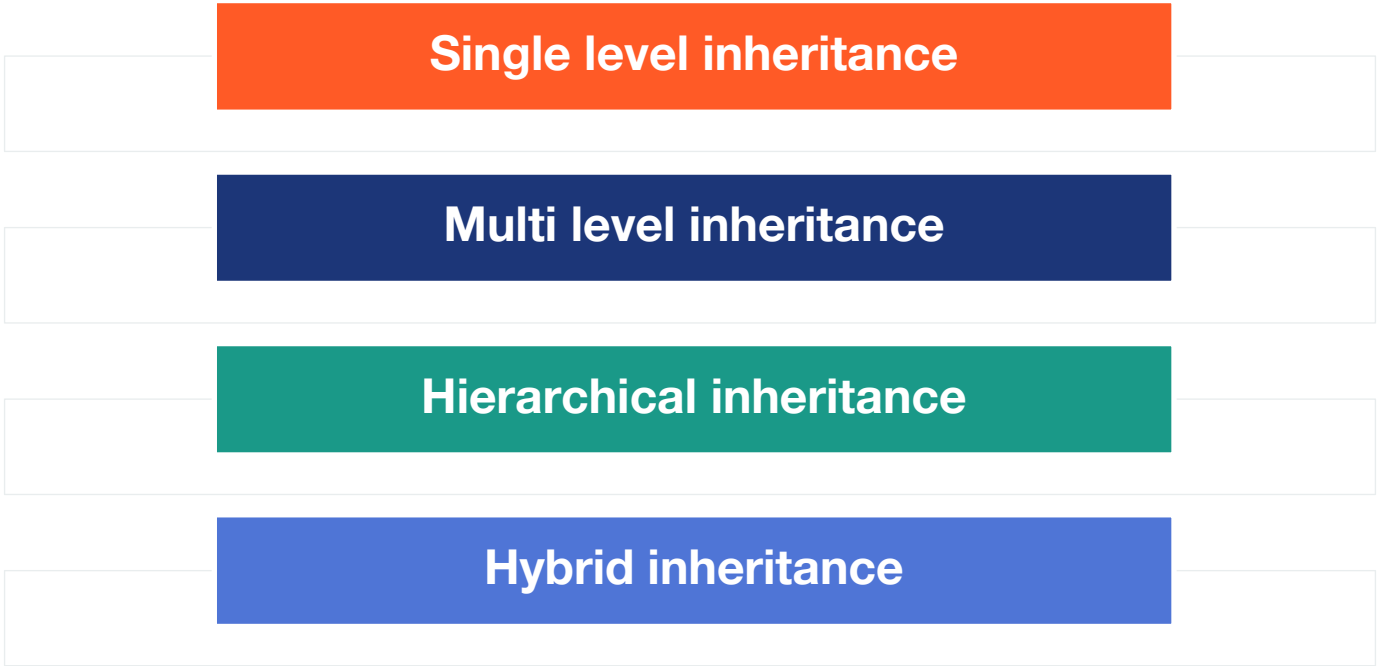
- Reusing or extending the functionalities and capabilities of the existing class in a new class is called Inheritance

- A class can inherit the features of a related class and add new features, as per the requirement

- Subclass is a class that inherits the data members and methods from another class. Additional data members and methods can be created to add more features to a subclass

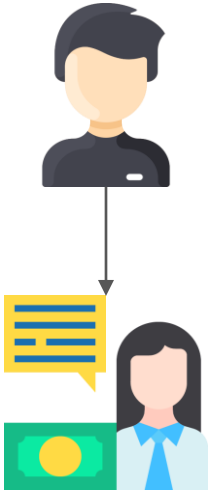
- Superclass is a class from which the subclass inherits. It can also be a subclass of another class

Types of Inheritance



Single Level Inheritance

- In single level inheritance, a single subclass derives the functionality of the existing superclass
- The following figure explains the single level inheritance



Single Level Inheritance (contd.)

- The following syntax shows how single level inheritance can be implemented

```
class Person
{
}
class Employee extends Person
{
}
```

Multi Level Inheritance

- In multilevel inheritance, a subclass inherits the properties of another subclass
- The following figure explains multi level inheritance

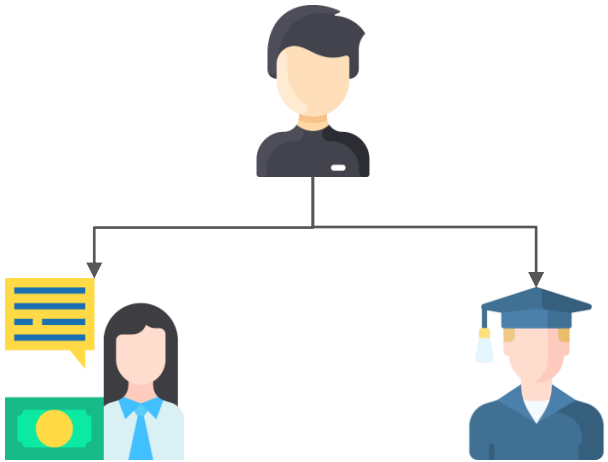
Multi Level Inheritance (contd.)

- The following syntax shows how multi level inheritance can be implemented

```
class Person
{
}
class Employee extends Person
{
}
class Manager extends Employee
{
}
```

Hierarchical Inheritance

- In hierarchical inheritance, one or more subclasses are derived from a single superclass
- The following figure explains the hierarchical Inheritance



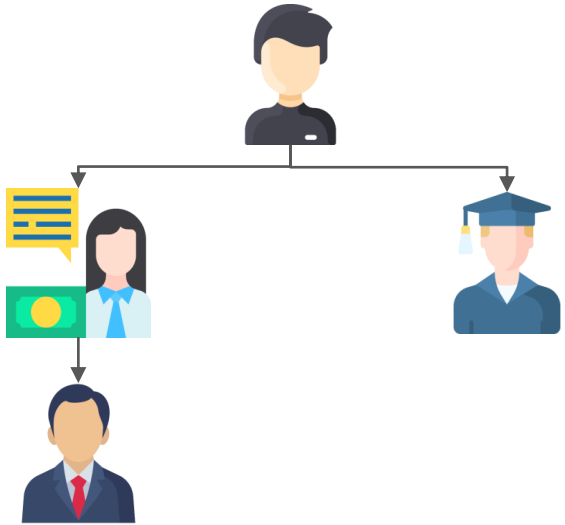
Hierarchical Inheritance (contd.)

- The following syntax shows how hierarchical inheritance can be implemented

```
class Person
{
}
class Employee extends Person
{
}
class Student extends Person
{
}
```

Hybrid Inheritance

- Hybrid inheritance is combination of one or two types of inheritance
- The following figure explains the hybrid inheritance

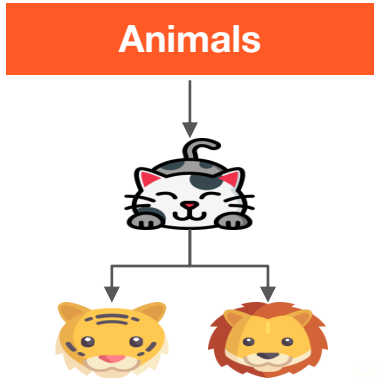


Hybrid Inheritance (contd.)

- The following syntax shows how to implement hybrid inheritance

```
class Person
{
}
class Employee extends Person
{
}
class Student extends Person
{
}
class Manager extends Employee
{
}
```

Interactive Demo



Look at the hierarchical classification of a few animal species shown in this image.

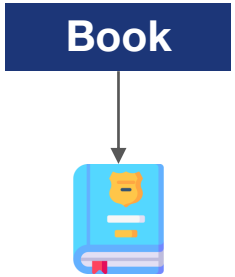
Task 1: Identify the common and unique attributes of these species.

Task 2: Identify the common and unique behaviors of these species.

Task 3: Implement the relationships between the species based on the common and unique attributes and behaviors identified by you.

Interactive Demo

Look at the hierarchical classification of a particular book.



Book class has the following attributes and behaviors:
Attributes: title, author, total number of pages, price, shipping charges
Behaviors: printing the book details
Paper book class shares the attributes and behaviors of the book class.

Implement this relationship based on the attributes and behaviors listed above.

Key Takeaways

- Relationship between classes
- Types of inheritance
- Differentiating between single level, multi level, hierarchical, and hybrid inheritance
- Implementing inheritance