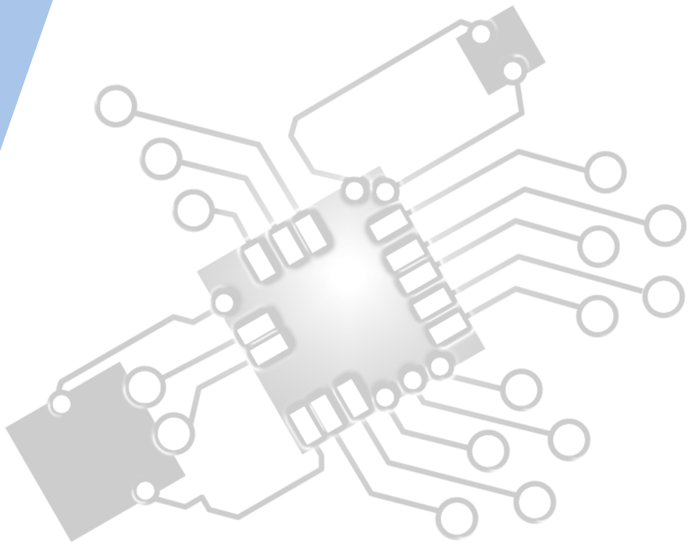
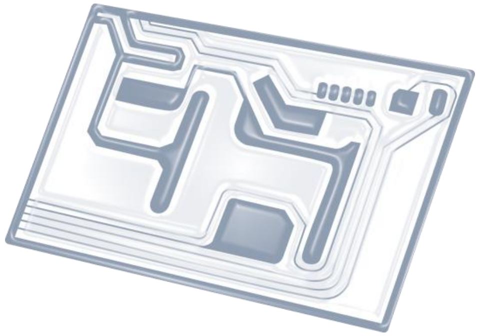
Features of OOP

IB Computer Science



Content developed by
Dartford Grammar School
Computer Science Department

HL Topics 1-7, D1-4



1: System design

2: Computer Organisation

3: Networks

4: Computational thinking

5: Abstract data structures

6: Resource management

7: Control

D: OOP

HL & SL D.2 Overview

D.2 Features of OOP

D.2.1 Define the term encapsulation

D.2.2 Define the term inheritance

D.2.3 Define the term polymorphism

D.2.4 Explain the advantages of encapsulation

D.2.5 Explain the advantages of inheritance

D.2.6 Explain the advantages of polymorphism

D.2.7 Describe the advantages of libraries of objects

D.2.8 Describe the disadvantages of OOP

D.2.9 Discuss the use of programming teams

D.2.10 Explain the advantages of modularity in program development

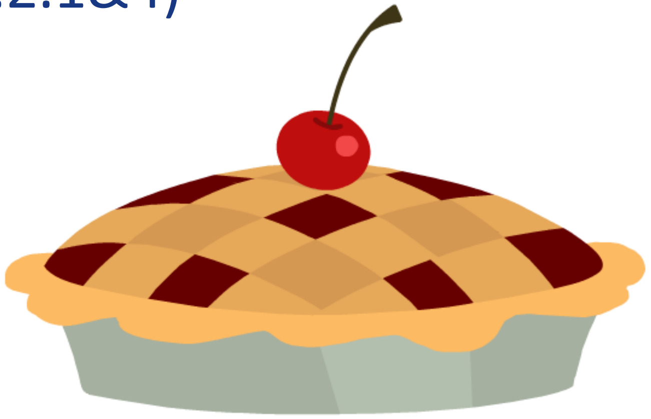
Topic D.2.2

Define the term: **inheritance**

"The meek will inherit the earth, but the inheritance tax will force them to sell it back to the rich."

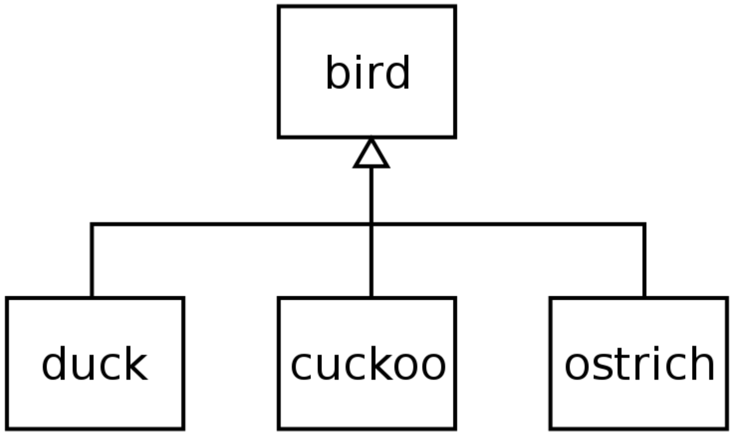
Four OOP fundamentals:

- Abstraction (*See* Topic 4.1.17-20)
- Polymorphism (*See* Topic D.2.3&6)
- Inheritance (*See* Topic D.2.2&5)
- Encapsulation (*See* Topic D.2.1&4)



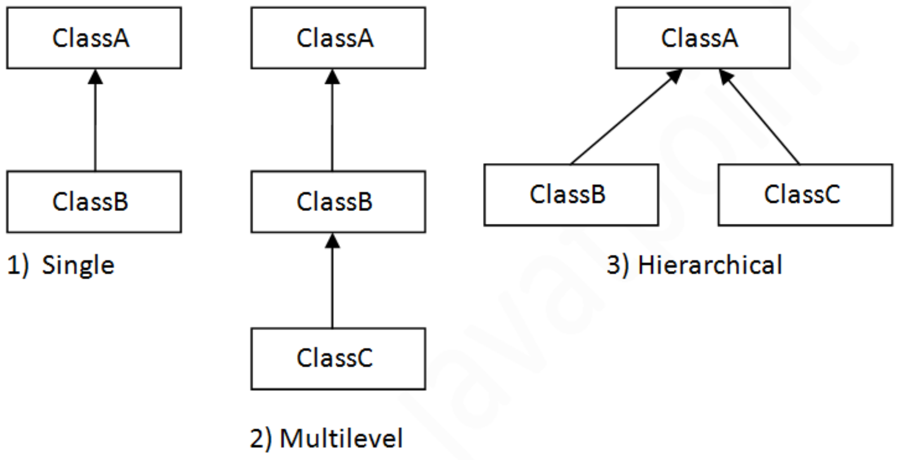
Definition: Inheritance

- Process whereby one object **inherits the properties** (states and behaviours) of another object (pairs called **super/sub** or **parent/child classes**)
- The Java keyword that implies inheritance is `extends`

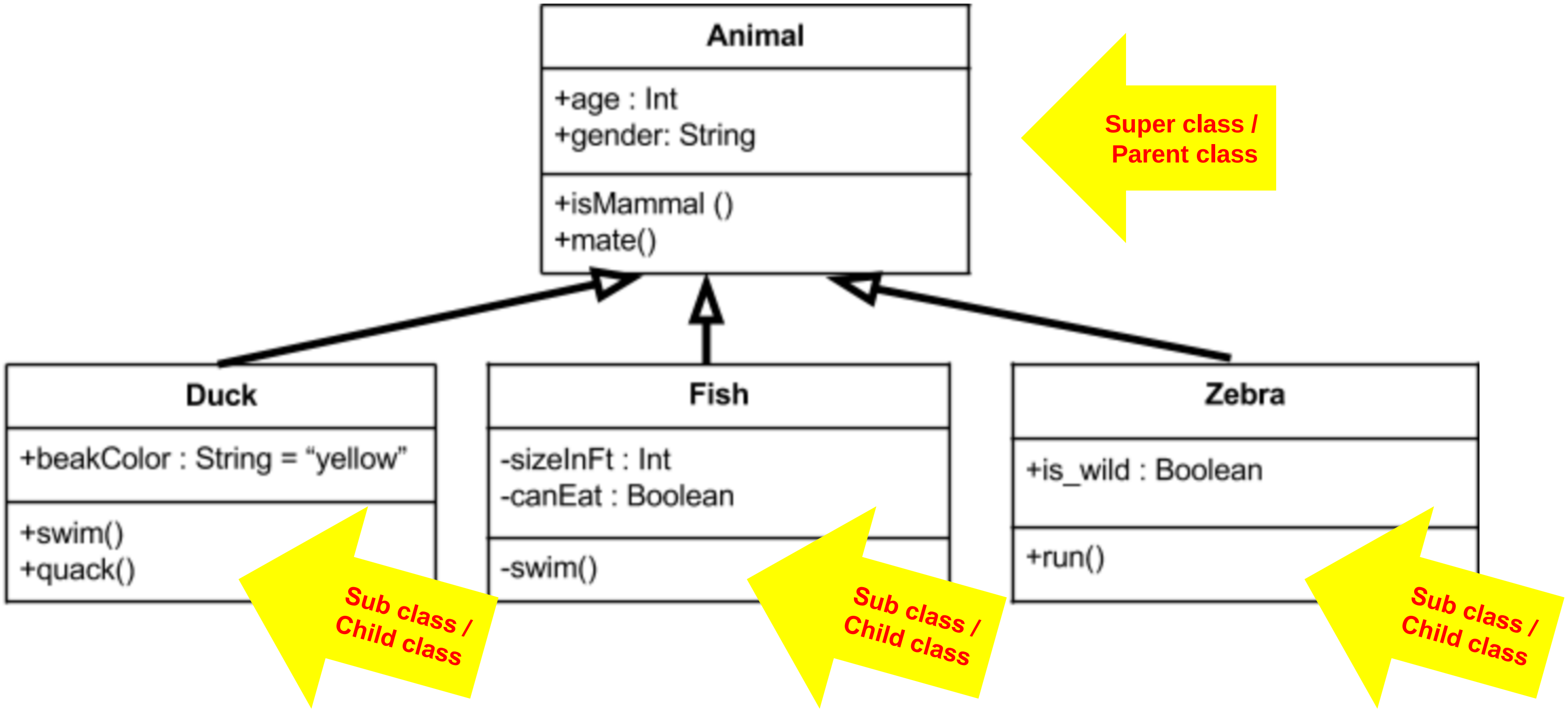


Key points

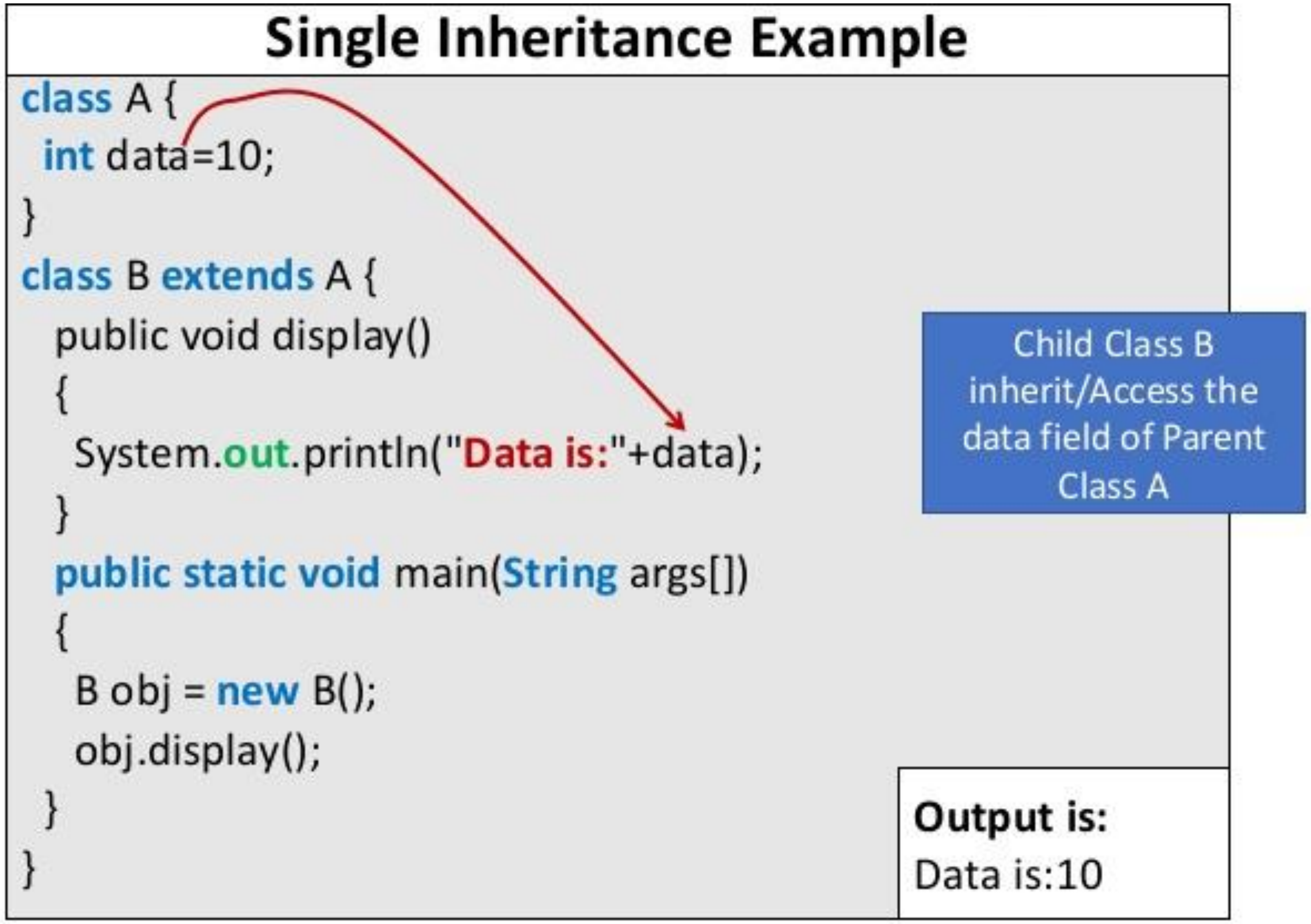
- In programming, inheritance is implemented by using the keyword **`extends`** in the **sub class** to connect it to its **super class**
- This is called an **'is-a'** relationship (*See* D.1.6)
- In Java, you can only inherit from **one class** at a time



Example: UML


Animal
+age : Int
+gender: String
+isMammal ()
+mate()
Super class / Parent class
Duck
+beakColor : String = “yellow”
+swim()
+quack()
Sub class / Child class
Fish
-sizeInFt : Int
-canEat : Boolean
-swim()
Sub class / Child class
Zebra
+is_wild : Boolean
+run()
Sub class / Child class

Example: Java


Single Inheritance Example

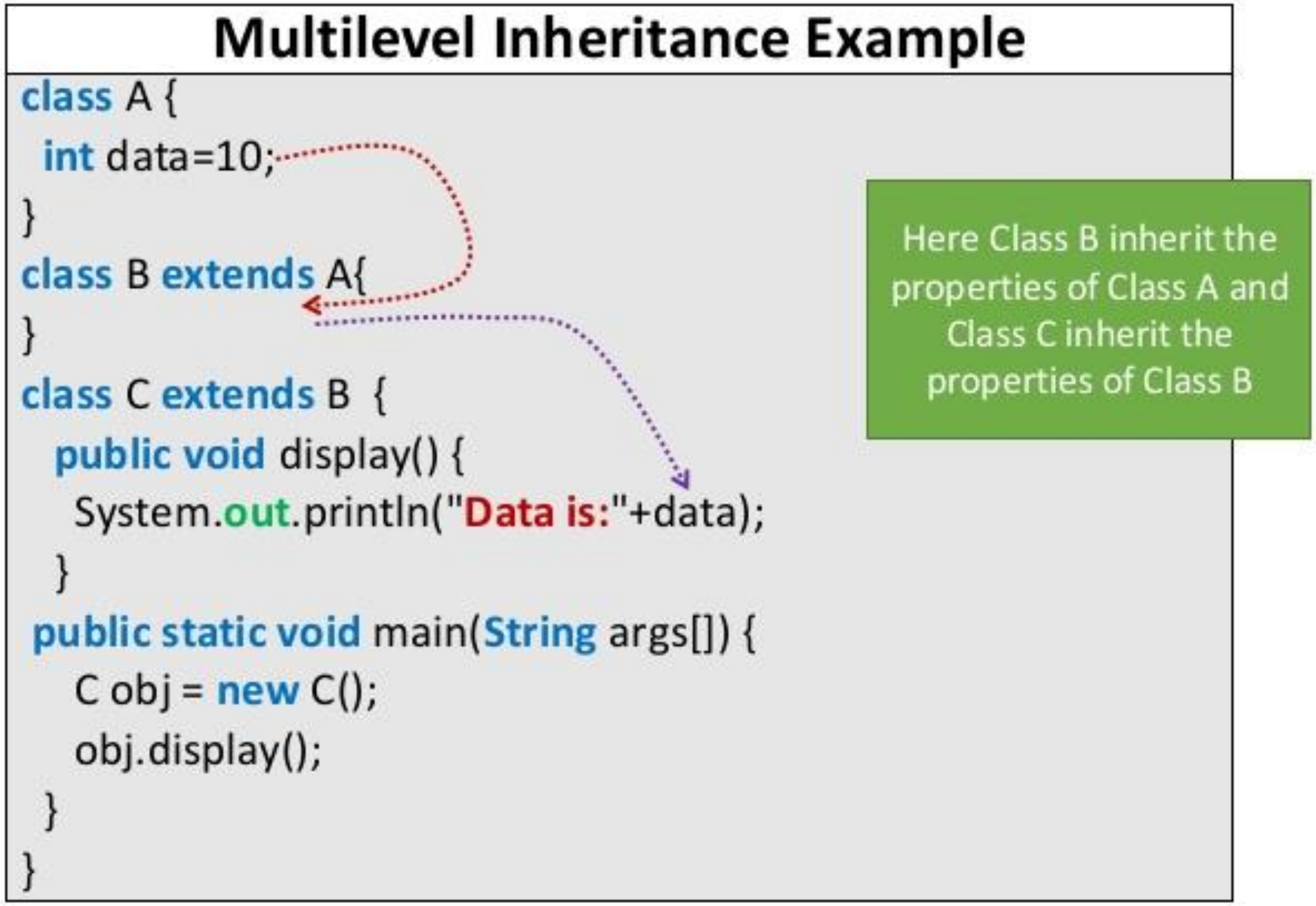
```
class A {
 int data=10;
}
class B extends A {
  public void display()
  {
   System.out.println("Data is:"+data);
  }
  public static void main(String args[])
  {
   B obj = new B();
   obj.display();
  }
}
```

Child Class B inherit/Access the data field of Parent Class A

Output is:

```
Data is:10
```

Example: Java



Multilevel Inheritance Example

```
class A {
  int data=10;
}
class B extends A{
}
class C extends B  {
   public void display() {
     System.out.println("Data is:"+data);
   }
 public static void main(String args[]) {
     C obj = new C();
     obj.display();
  }
}
```

Here Class B inherit the properties of Class A and Class C inherit the properties of Class B