THE UNIVERSITY OF MELBOURNE

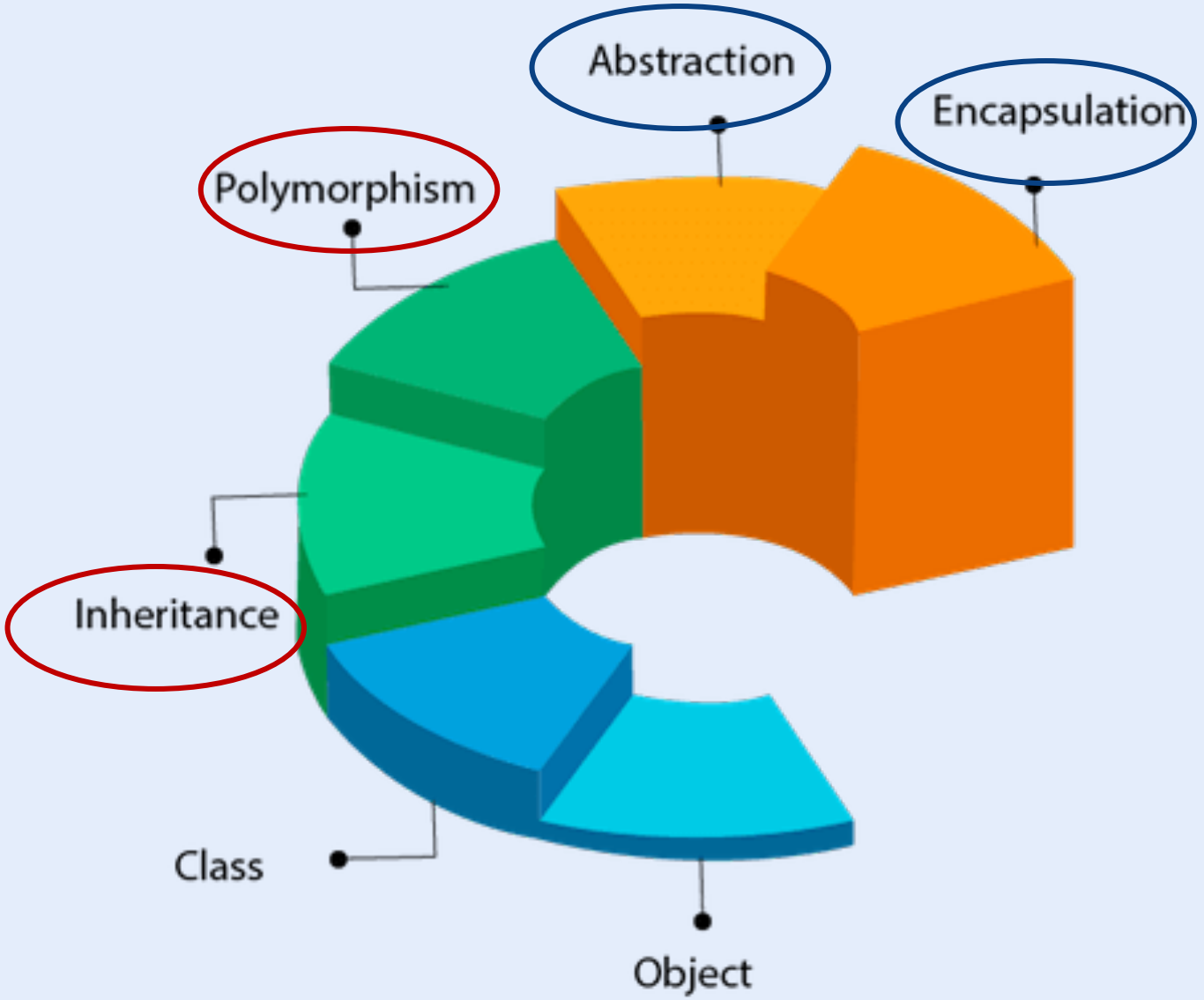
Lab Week9

COMP90041 Programming and software development

Zhe(Zoe) Wang

github: https://github.com/Zoeewang/COMP90041-2020-sem1-tutorial

OOPs (Object-Oriented Programming System)



inheritance

subclass / child class

inheritance allows a derived class to be defined by specifying only how it differs from base class superclass / parent class

form: extends BaseClass{ …. }

```
public class Person {
    private int age;
    private String name;}
```

```
public class LostPerson extends Person {
    private String location;
    private int date;

}
```

LostPerson class inherits all the instance variables and methods of the Person class….and adds its own!

No need to mention inherited instance variables and methods

every instance of the derived class is also an instance of the base class (every lostperson is a person)

super Constructor

Constructors are not inherited, cannot be overridden !

Constructor chaining: derived class constructor must invoke base class constructor first. unless default constructors are in place for both classes.

form: super(arguments…)

```
public Person(int age, String name) {
    this.age = age;
    this.name = name;
}
```

```
public LostPerson(int age, String name, String location, int date) {
    super(age, name);
    this.location = location;
    this.date = date;
}
```

Overriding

If a class defines a method with same signature as an ancestor, its definition overrides the ancestor's

In Person:

```
public String toString(){
    return "name: " + name + " age: " + age;
}
```

In LostPerson:

```
public String toString(){
    return "name: " + getName() + " age: " + getAge() + " location: "
            +location + " date: " + date;
}
```

Use overridden methods

inside a method, use super.methodName(args…) to invoke the overridden methods

```
public String toString(){
    return "name: " + getName() + " age: " + getAge() + " location: "
            +location + " date: " + date;
}
```

```
public String toString(){
    return super.toString() + " location: " + location + " date: " + date;
}
```

Method Overriding vs Overloading (Polymorphism)

Overriding

a subclass can supply its own implementation for a method that also exists in the superclass

In Person:

```
public void greet(String name){
    System.out.println("hello"+ name);
}
```

In LostPerson:

```
public void greet(String name){
    System.out.println("Find" + name);
}
```

Overloading

two methods have same name but have different signatures

```
public void greet(String name){
    System.out.println("hello"+ name);
}

public void greet(){
    System.out.println("hello");
}
```

Q1

The class **Figure** is the base class, You should add only Rectangle and Triangle classes derived from Figure.
Each class has stubs for methods erase and draw. Each of these methods outputs a message telling the name of the class and what method has been called.

The method center calls the erase and draw methods to erase and redraw the figure at the center.

Copy constructor

a constructor that takes one argument of the same type as the object being constructed

- just make the new object an exact copy of the input argument

```
public Dog(Dog orig){
    this.age = orig.age;
    this.name = orig.name;
}
```

Late Binding

Person p1 = **new** **LostPerson**(...)

Declared type (what methods available)

actual type (which method implementation will be used)

```
Person p1 = new LostPerson(12,"bob","mel",20200502);
System.out.println(p1);
```

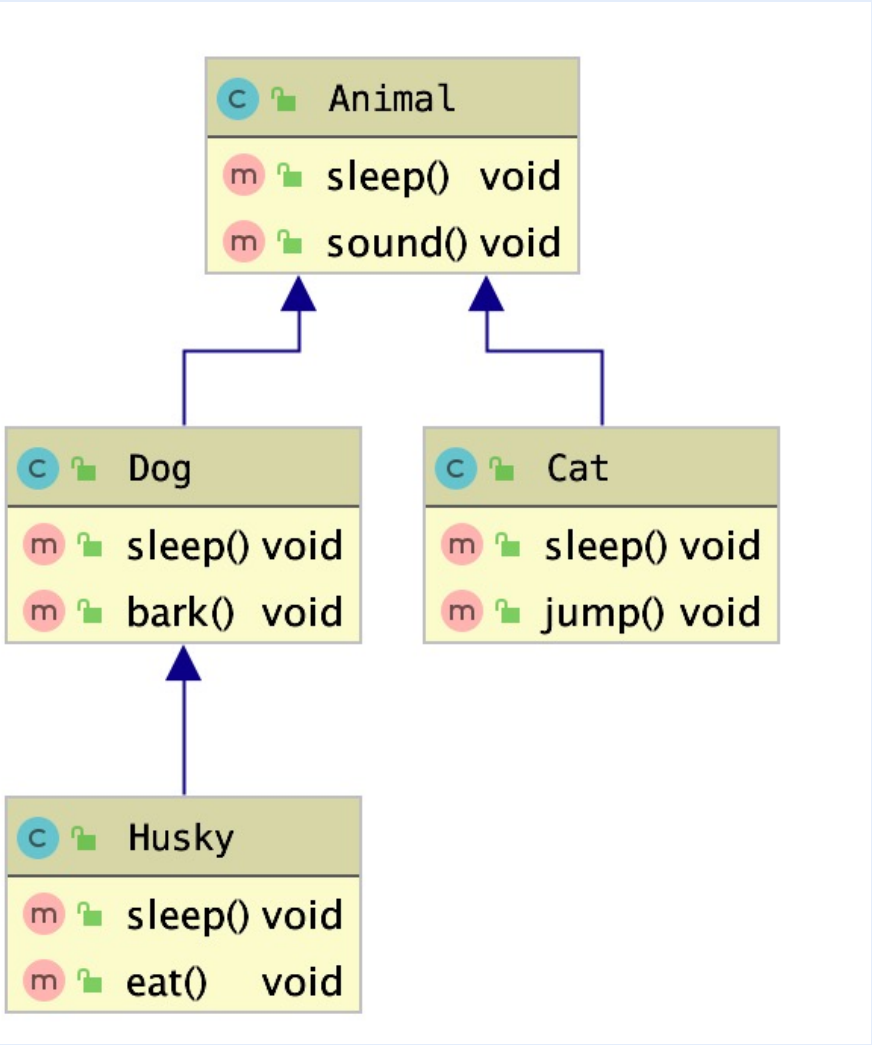
which toString method is used??
LostPerson / Person ?

Late Binding

```
Person p1 = new LostPerson(...)
```

Declared type (what methods available)

actual type (which method implementation will be used)




```
Animal a1 = new Dog();
Animal a2 = new Cat();
Dog d1 = new Dog();
Dog d2 = new Husky();
```

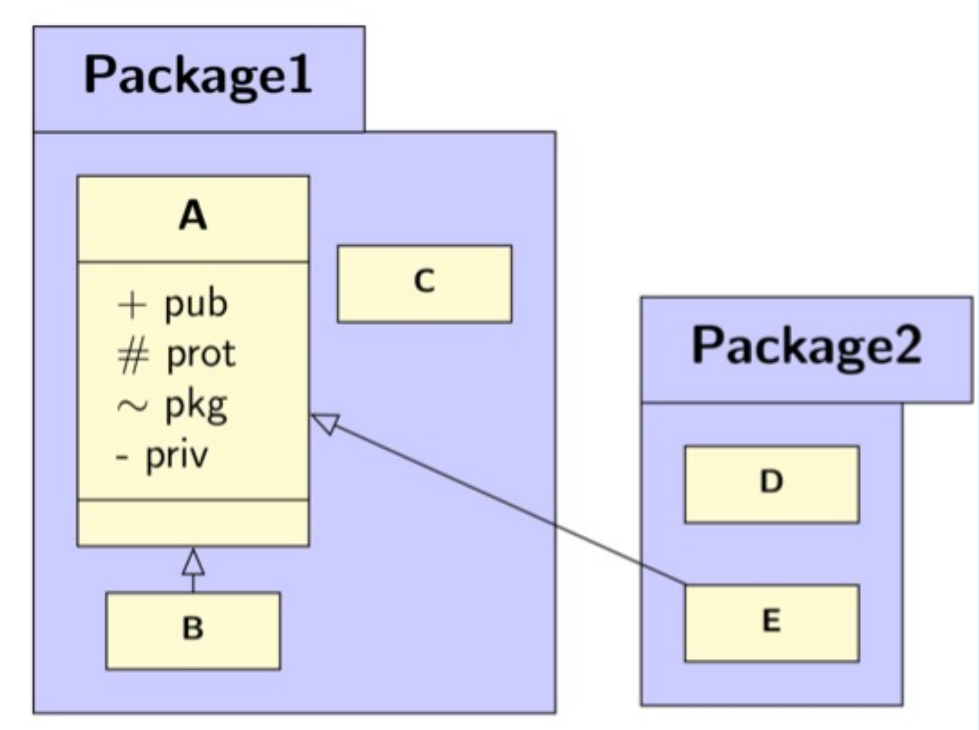
//which statements are wrong/invalid?

1. a1.sleep();
2. a1.bark();
3. a2.sleep();
4. a2.sound();
5. d1.bark();
6. d2.eat();

Visibility

private < default(package) < protected < public

(package + subclass)



A sees pub, prot, pkg, priv

B sees pub, prot, pkg

C sees pub, prot, pkg

D sees pub

E sees pub, prot

THE UNIVERSITY OF MELBOURNE

Thank you