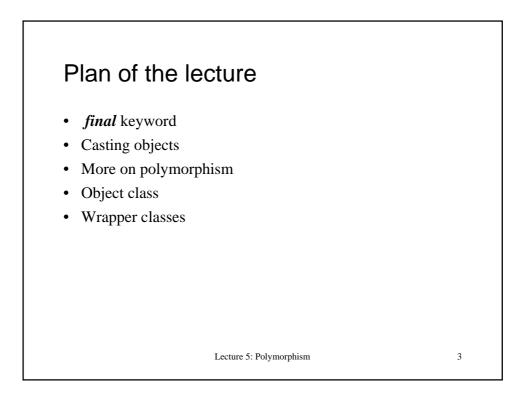
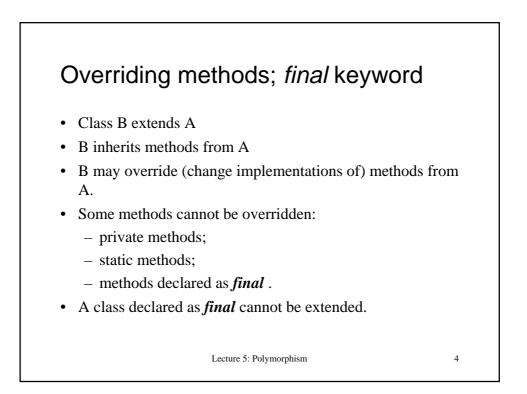
G51PRG: Introduction to Programming Second semester Lecture 5

Natasha Alechina School of Computer Science & IT nza@cs.nott.ac.uk

<section-header><section-header><section-header><text><list-item><list-item><list-item>







• You already saw *final* used in declarations of variables which cannot be changed in the program (e.g. constants).

final double pi = 3.14;

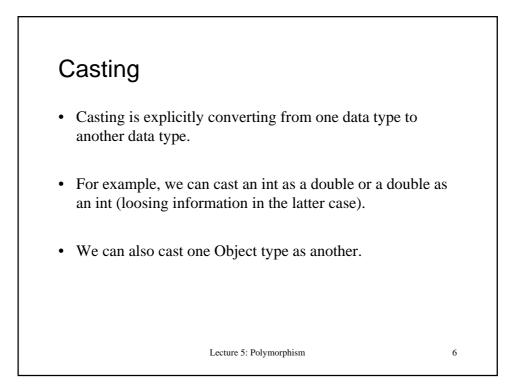
```
pi = 4.0;
```

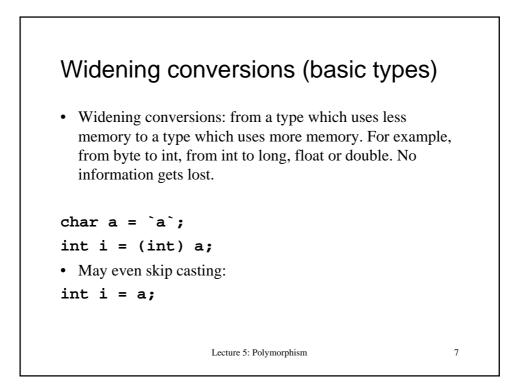
will cause a compiler error.

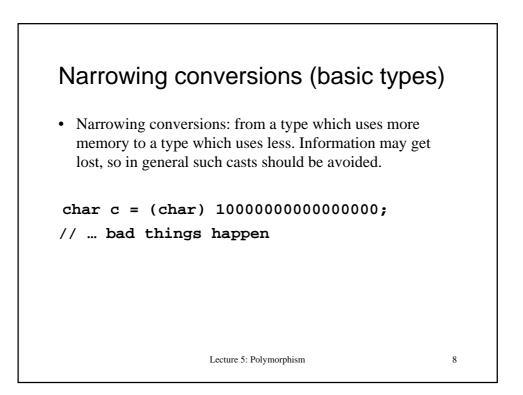
- Another use of *final* is to prevent overriding methods/changing class definitions.
- Reasons for doing this: preventing errors, efficiency and (later in the lecture) security.

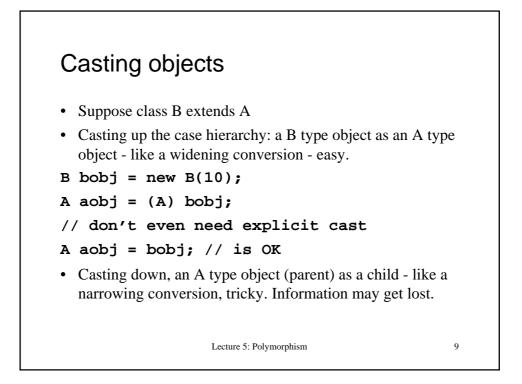
Lecture 5: Polymorphism

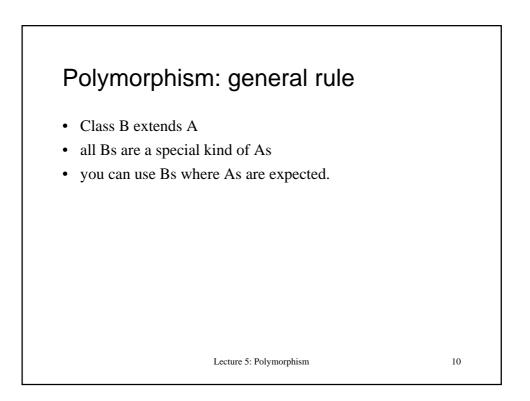
5

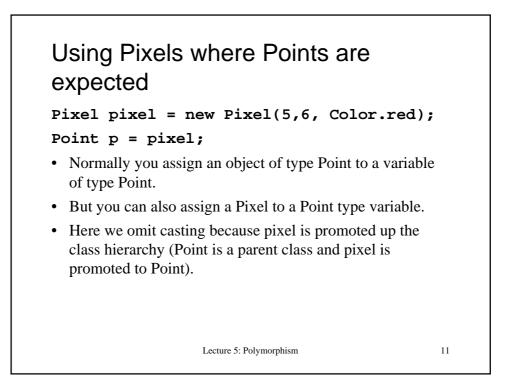


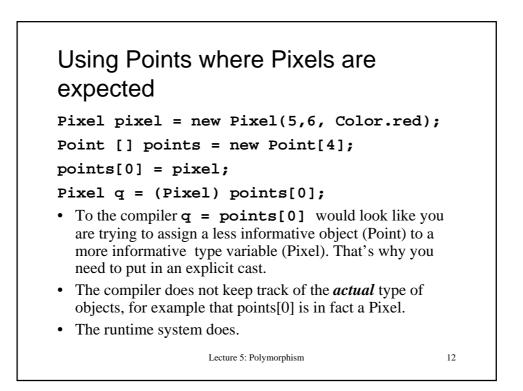












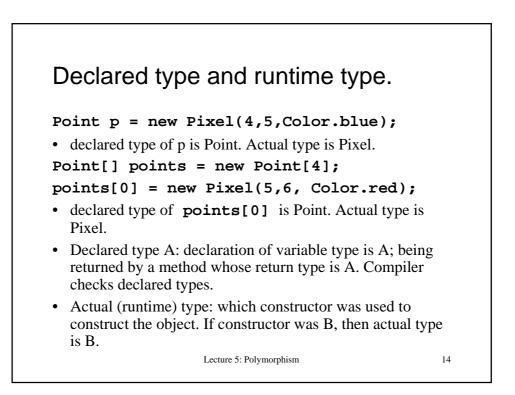
Runtime errors

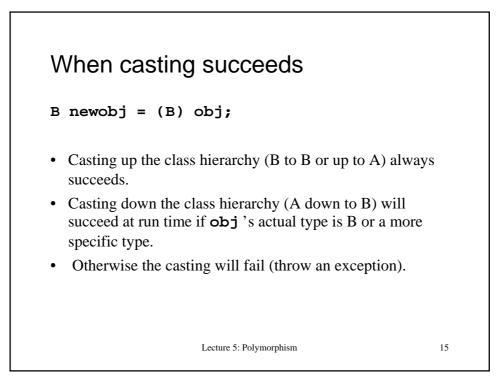
```
Point [] points = new Point[4];
points[0] = new Point(5,6);
Pixel q = (Pixel) points[0];
```

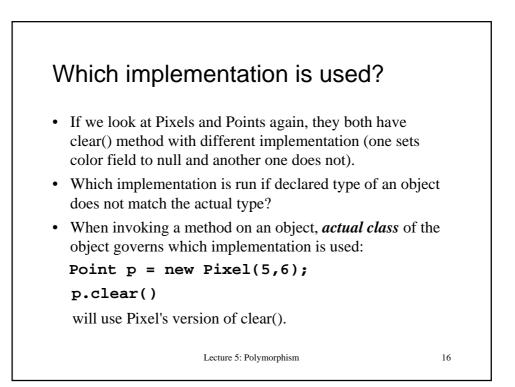
- This will pass the compiler, same as the code before.
- The runtime system will throw ClassCastNotSupportedException.
- Casting "real" Pixel to a Pixel will work.
- Casting a Point to a Pixel will not work.

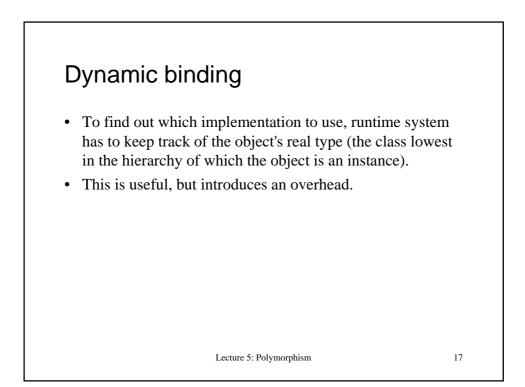
Lecture 5: Polymorphism

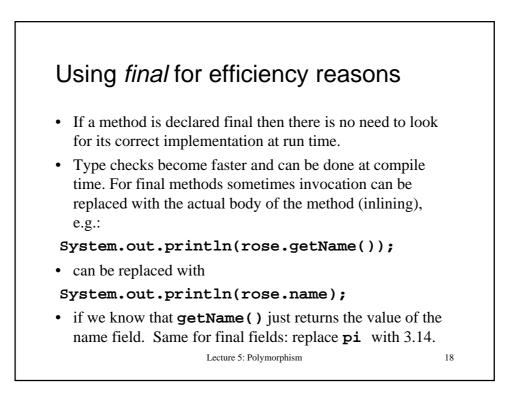
13

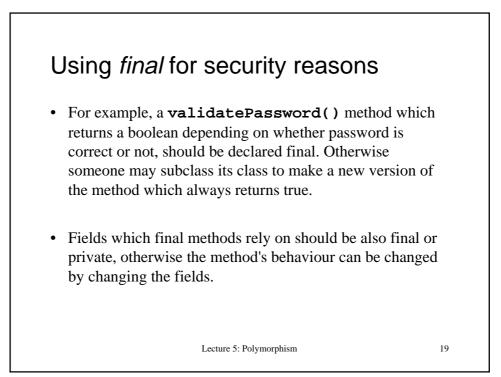


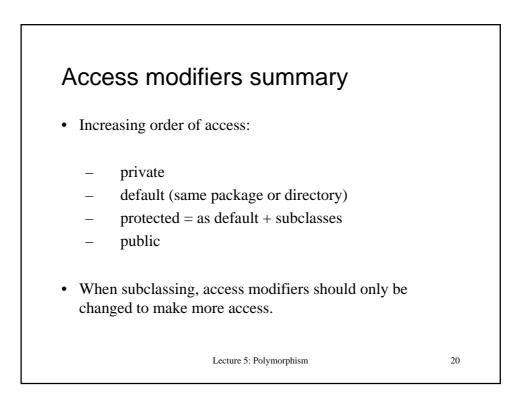


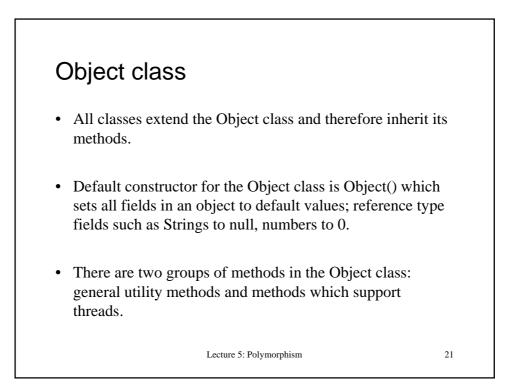


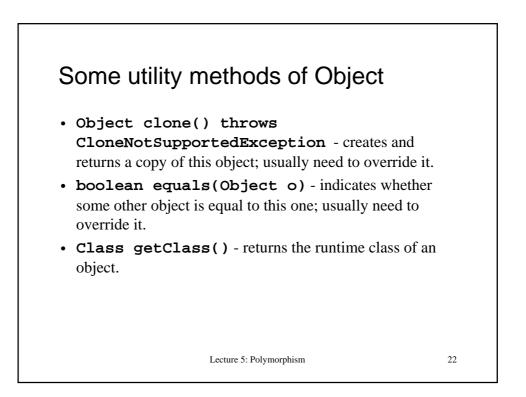


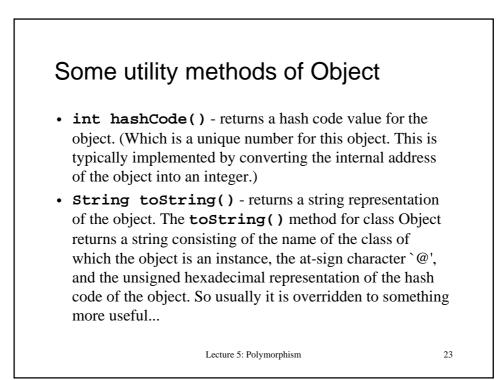












Example

class Person {
 String name;
 int age;
 Person(String s, int n) {
 this.name = new String(s);
 this.age = n;
 }
 public Object clone() {
 return new Person(this.name, this.age);
 }}

