

Introduction to Programming

with Java, for Beginners

Casting

Casts with Reference Types

```
public class Person {
    protected String name;

    //Constructor
    public Person(String s){
        name = s;
    }

    //Methods
    public String getName(){
        return name;
    }

    public String toString(){
        return "Person: " +
            getName();
    }
}
ESE112
```

```
public class Student extends Person{
    private int yearOfGraduation;

    public Student(String n, int year){
        super(n);
        yearOfGraduation = year;
    }

    public String toString(){
        return "Student: " + getName();
    }

    public int getYearOfGraduation(){
        return yearOfGraduation;
    }
}
```

2

Casts with Reference Types (contd..)

```
> Person p = new Student("Harry", 2011);
> p.getName()
"Harry"
> p.getYearOfGraduation()
```

Error no getYearOfGraduation() in Person

- Illegal because "p" can only execute information from Person class
 - > Person class has no info about getYearOfGraduation()
- The solution here is to use type-casting.

ESE112

3/15

Casts with Reference Types (contd..)

- If, for some reason, you happen to know that "p" does in fact refer to a Student, you can use the **type cast**
 - > Cast tells the computer to treat "p" as if it were actually of type Student
 - > i.e. ((Student)p). getYearOfGraduation();
- To make sure that p is referring to Student use **instanceof** operator
 - > If is not then you will get a runtime error

```
if(v instanceof Student ){
    ((Student)p). getYearOfGraduation()
}
```

ESE112

4