Inheritance in Java

By the end of this video you will be able to…

- Explain the value of inheritance as a feature in object oriented programming languages

Potential Problem…

Fully written Person class now needs to handle:

1. Students
2. Faculty

```
public class Person
{

 private String name;
 …



}
```

Potential Solution 1

```
public class Person
{
 private String name;




 …
}
```

Potential Solution 1

```
public class Person
{
 private String name;
 private boolean student;




 …
}
```

Potential Solution 1

```
public class Person
{
 private String name;
 private boolean student;
 public Person(boolean s)
 {
  this.student = s;
 }
 …
}
```

Potential Solution 1

```
public class Person
{
 private String name;
 private boolean student;
 public Person(boolean s)
 {
  this.student = s;
 }
 …
}
```

Now in every method, I can just do this:

```
if (student)
   // code for students
else
   // code for faculty
```

Potential Solution 1 – Problems…

```
public class Person
{
 private String name;
 private boolean student;
 private boolean graduate;
 private boolean fullTime;

 …
}
```

Each method becomes:

```
if (student)
   if (graduate && fullTime)
      // some code
   else if (!graduate)
      // more code
// Ack!!
```

Potential Solution 1 – Problems…

```
public class Person
{
 private String name;
 private boolean student;
 private boolean graduate;
 private boolean fullTime;


 …
}
```

Each method becomes:

```
if (student)
   if (graduate && fullTime)
      …
   el…
      /…
// Ach…
```

Spaghetti Code

Potential Solution 2

```
public class Student
{
 private String name;

 …



}
```

```
public class Faculty
{
 private String name;

 …



}
```

Potential Solution 2 – Problems…

```
public class Student
{
 private String name;

 …



}
```

```
public class Faculty
{
 private String name;

 …



}
```

Potential Solution 2 – Problems…

```
public class Student
{
 private String firstName;
 private String lastName;

 …


}
```

```
public class Faculty
{
 private String name;

 …


}
```

Potential Solution 2 – Problems…

```
public class Student
{
 private String firstName;
 private String lastName;

 …


}
```

```
public class Faculty
{
 private String name;


 …
```

Hard Keep Common Code Consistent

Potential Solution 2 – Problems…

```
public class Student
{
 private String name;
 …
}
```

```
public class Faculty
{
 private String name;
 …
}
```

```
// in main
Person persons[];
```

Potential Solution 2 – Problems…

```
public class Student
{
 private String name;
 …
}
```

```
public class Faculty
{
 private String name;
 …
}
```

```
// in main
Person persons[];
```

Potential Solution 2 – Problems…

```
public class Student
{
 private String name;
 …
}
```

```
public class Faculty
{
 private String name;
 …
}
```

```
// in main
~~Person persons[];~~
Student students[];
Faculty faculty[];
```

No Clean Way Single Array of Everyone

What do we want then?

1. Keep common behavior in one class

What do we want then?

1. Keep common behavior in one class
2. Split different behavior into separate classes

What do we want then?

1. Keep common behavior in one class
2. Split different behavior into separate classes
3. Keep all of the objects in a single data structure

Yay Inheritance!!!!