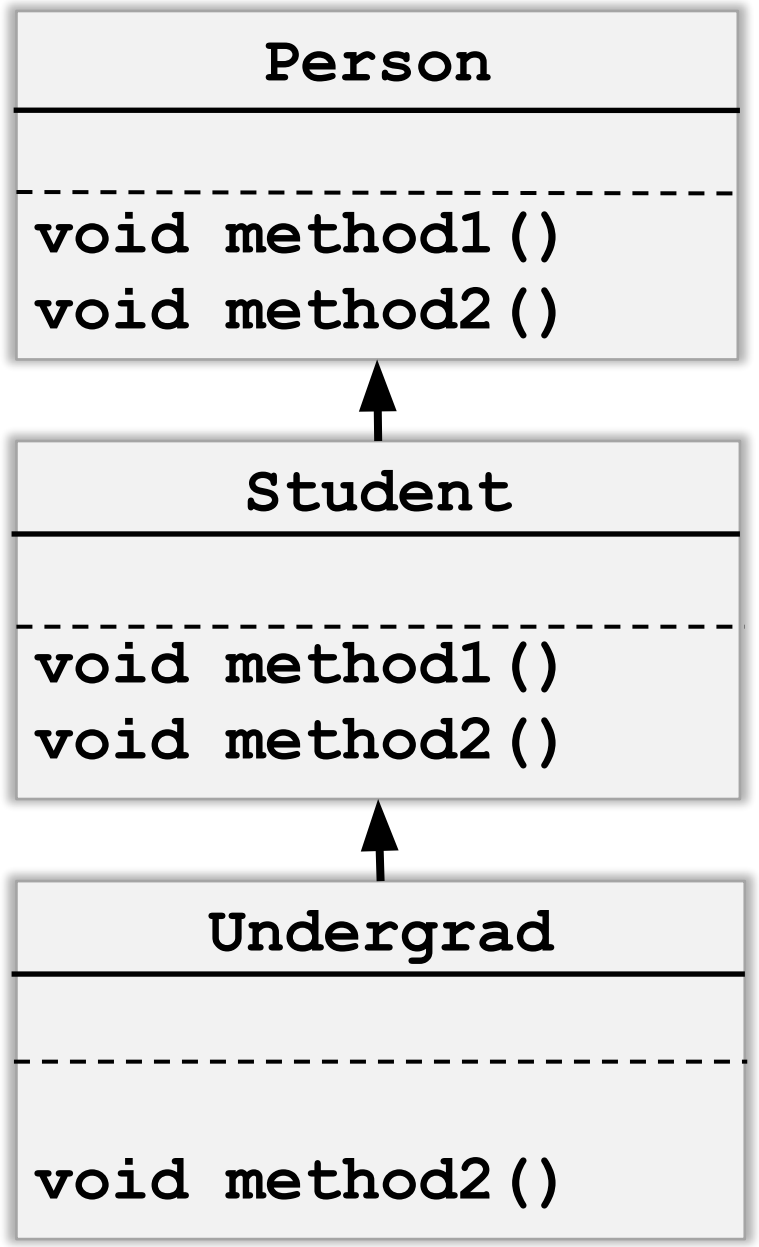
Polymorphism: Concept Challenge

Concept Challenge

- **Pause** Try to solve the problem yourself
- **Discuss** with other learners (if you can)
- **Watch** the UCSD learners video
- **Confirm** your understanding with our explanation


Person
void method1()
void method2()
Student
void method1()
void method2()
Undergrad
void method2()

```
public class Person {
 public void method1() {
  System.out.print("Person 1 ");
 }
 public void method2() {
  System.out.print("Person 2 ");
 }
}
public class Student extends Person {
 public void method1() {
  System.out.print("Student 1 ");
  super.method1();
  method2();
 }
 public void method2() {
  System.out.print("Student 2 ");
 }
}
public class Undergrad extends Student {
  public void method2() {
   System.out.print("Undergrad 2 ");
  }
}
```

```
Person u = new Undergrad();
u.method1();
```

Start IVQ

```
public class Person {
 public void method1() {
  System.out.print("Person 1 ");
 }
 public void method2() {
  System.out.print("Person 2 ");
 }
}
public class Student extends Person {
 public void method1() {
  System.out.print("Student 1 ");
  super.method1();
  method2();
 }
 public void method2() {
  System.out.print("Student 2 ");
 }
}
public class Undergrad extends Student {
  public void method2() {
   System.out.print("Undergrad 2 ");
  }
}
```

```
Person u = new Undergrad();
u.method1();
```

<IVQ placeholder> E is correct.
What gets printed?

A) Compiler error
B) Run time error
C) Infinite execution
D) Student 1 / Person 1 / Student 2
E) Student 1 / Person 1 / Undergrad 2

End IVQ / Start Discussion

```
public class Person {
 public void method1() {
  System.out.print("Person 1 ");
 }
 public void method2() {
  System.out.print("Person 2 ");
 }
}
public class Student extends Person {
 public void method1() {
  System.out.print("Student 1 ");
  super.method1();
  method2();
 }
 public void method2() {
  System.out.print("Student 2 ");
 }
}
public class Undergrad extends Student {
  public void method2() {
   System.out.print("Undergrad 2 ");
  }
}
```

```
Person u = new Undergrad();
u.method1();
```

```
public class Person {
 public void method1() {
  System.out.print("Person 1 ");
 }
 public void method2() {
  System.out.print("Person 2 ");
 }
}
public class Student extends Person {
 public void method1() {
  System.out.print("Student 1 ");
  super.method1();
  method2();
 }
 public void method2() {
  System.out.print("Student 2 ");
 }
}
public class Undergrad extends Student {
  public void method2() {
   System.out.print("Undergrad 2 ");
  }
}
```

```
Person u = new Undergrad();
u.method1();
```

```
public class Person {
 public void method1() {
  System.out.print("Person 1 ");
 }
 public void method2() {
  System.out.print("Person 2 ");
 }
}
public class Student extends Person {
 public void method1() {
  System.out.print("Student 1 ");
  super.method1();
  method2();
 }
 public void method2() {
  System.out.print("Student 2 ");
 }
}
public class Undergrad extends Student {
  public void method2() {
   System.out.print("Undergrad 2 ");
  }
}
```

```
Person u = new Undergrad();
u.method1();
```

```
public class Person {
 public void method1() {
  System.out.print("Person 1 ");
 }
 public void method2() {
  System.out.print("Person 2 ");
 }
}
public class Student extends Person {
 public void method1() {
  System.out.print("Student 1 ");
  super.method1();
  method2();
 }
 public void method2() {
  System.out.print("Student 2 ");
 }
}
public class Undergrad extends Student {
  public void method2() {
   System.out.print("Undergrad 2 ");
  }
}
```

```
Person u = new Undergrad();
u.method1();
```

```
public class Person {
 public void method1() {
  System.out.print("Person 1 ");
 }
 public void method2() {
  System.out.print("Person 2 ");
 }
}
public class Student extends Person {
 public void method1() {
  System.out.print("Student 1 ");
  super.method1();
  method2();
 }
 public void method2() {
  System.out.print("Student 2 ");
 }
}
public class Undergrad extends Student {
  public void method2() {
   System.out.print("Undergrad 2 ");
  }
}
```

```
Person u = new Undergrad();
u.method1();
```

```
public class Person {
 public void method1() {
  System.out.print("Person 1 ");
 }
 public void method2() {
  System.out.print("Person 2 ");
 }
}
public class Student extends Person {
 public void method1() {
  System.out.print("Student 1 ");
  super.method1();
  method2();
 }
 public void method2() {
  System.out.print("Student 2 ");
 }
}
public class Undergrad extends Student {
  public void method2() {
   System.out.print("Undergrad 2 ");
  }
}
```

```
Person u = new Undergrad();
u.method1();
```

Output so far:

Student 1

```
public class Person {
 public void method1() {
  System.out.print("Person 1 ");
 }
 public void method2() {
  System.out.print("Person 2 ");
 }
}
public class Student extends Person {
 public void method1() {
  System.out.print("Student 1 ");
  super.method1();
  method2();
 }
 public void method2() {
  System.out.print("Student 2 ");
 }
}
public class Undergrad extends Student {
  public void method2() {
   System.out.print("Undergrad 2 ");
  }
}
```

```
Person u = new Undergrad();
u.method1();
```

Output so far:

```
Student 1
```

```
public class Person {
 public void method1() {
  System.out.print("Person 1 ");
 }
 public void method2() {
  System.out.print("Person 2 ");
 }
}
public class Student extends Person {
 public void method1() {
  System.out.print("Student 1 ");
  super.method1();
  method2();
 }
 public void method2() {
  System.out.print("Student 2 ");
 }
}
public class Undergrad extends Student {
  public void method2() {
   System.out.print("Undergrad 2 ");
  }
}
```

```
Person u = new Undergrad();
u.method1();
```

Output so far:

```
Student 1
```

super of what?

```
public class Person {
 public void method1() {
  System.out.print("Person 1 ");
 }
 public void method2() {
  System.out.print("Person 2 ");
 }
}
public class Student extends Person {
 public void method1() {
  System.out.print("Student 1 ");
  super.method1();
  method2();
 }
 public void method2() {
  System.out.print("Student 2 ");
 }
}
public class Undergrad extends Student {
  public void method2() {
   System.out.print("Undergrad 2 ");
  }
}
```

```
Person u = new Undergrad();
u.method1();
```

Output so far:

```
Student 1
```

Static binding!

```
public class Person {
 public void method1() {
  System.out.print("Person 1 ");
 }
 public void method2() {
  System.out.print("Person 2 ");
 }
}
public class Student extends Person {
 public void method1() {
  System.out.print("Student 1 ");
  super.method1();
  method2();
 }
 public void method2() {
  System.out.print("Student 2 ");
 }
}
public class Undergrad extends Student {
  public void method2() {
   System.out.print("Undergrad 2 ");
  }
}
```

```
Person u = new Undergrad();
u.method1();
```

Output so far:

```
Student 1
```

```
public class Person {
 public void method1() {
  System.out.print("Person 1 ");
 }
 public void method2() {
  System.out.print("Person 2 ");
 }
}
public class Student extends Person {
 public void method1() {
  System.out.print("Student 1 ");
  super.method1();
  method2();
 }
 public void method2() {
  System.out.print("Student 2 ");
 }
}
public class Undergrad extends Student {
  public void method2() {
   System.out.print("Undergrad 2 ");
  }
}
```

```
Person u = new Undergrad();
u.method1();
```

Output so far:

```
Student 1
```

```
public class Person {
 public void method1() {
  System.out.print("Person 1 ");
 }
 public void method2() {
  System.out.print("Person 2 ");
 }
}
public class Student extends Person {
 public void method1() {
  System.out.print("Student 1 ");
  super.method1();
  method2();
 }
 public void method2() {
  System.out.print("Student 2 ");
 }
}
public class Undergrad extends Student {
  public void method2() {
   System.out.print("Undergrad 2 ");
  }
}
```

```
Person u = new Undergrad();
u.method1();
```

Output so far:

Student 1 Person 1

```
public class Person {
 public void method1() {
  System.out.print("Person 1 ");
 }
 public void method2() {
  System.out.print("Person 2 ");
 }
}
public class Student extends Person {
 public void method1() {
  System.out.print("Student 1 ");
  super.method1();
  method2();
 }
 public void method2() {
  System.out.print("Student 2 ");
 }
}
public class Undergrad extends Student {
  public void method2() {
   System.out.print("Undergrad 2 ");
  }
}
```

```
Person u = new Undergrad();
u.method1();
```

Output so far:

```
Student 1 Person 1
```

```
public class Person {
 public void method1() {
  System.out.print("Person 1 ");
 }
 public void method2() {
  System.out.print("Person 2 ");
 }
}
public class Student extends Person {
 public void method1() {
  System.out.print("Student 1 ");
  super.method1();
  method2();
 }
 public void method2() {
  System.out.print("Student 2 ");
 }
}
public class Undergrad extends Student {
  public void method2() {
   System.out.print("Undergrad 2 ");
  }
}
```

```
Person u = new Undergrad();
u.method1();
```

Output so far:

```
Student 1 Person 1
```

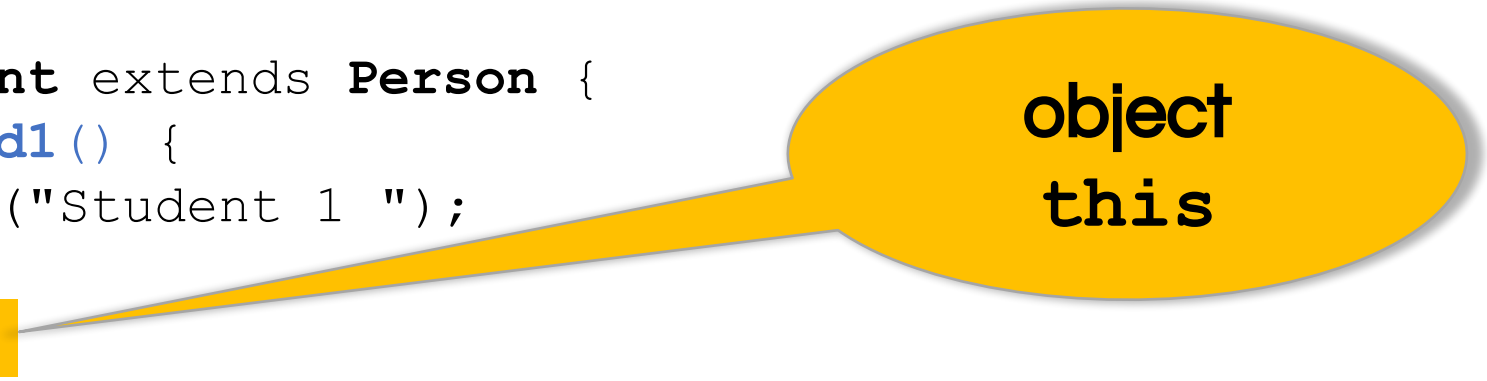
which method2()?

```
public class Person {
 public void method1() {
  System.out.print("Person 1 ");
 }
 public void method2() {
  System.out.print("Person 2 ");
 }
}
public class Student extends Person {
 public void method1() {
  System.out.print("Student 1 ");
  super.method1();
  this.method2();
 }
 public void method2() {
  System.out.print("Student 2 ");
 }
}
public class Undergrad extends Student {
  public void method2() {
   System.out.print("Undergrad 2 ");
  }
}
```

```
Person u = new Undergrad();
u.method1();
```

Output so far:

```
Student 1 Person 1
```



```
public class Person {
 public void method1() {
  System.out.print("Person 1 ");
 }
 public void method2() {
  System.out.print("Person 2 ");
 }
}
public class Student extends Person {
 public void method1() {
  System.out.print("Student 1 ");
  super.method1();
  this.method2();
 }
 public void method2() {
  System.out.print("Student 2 ");
 }
}
public class Undergrad extends Student {
  public void method2() {
   System.out.print("Undergrad 2 ");
  }
}
```

```
Person u = new Undergrad();
u.method1();
```

Output so far:

```
Student 1 Person 1
```

type of object at runtime

```
public class Person {
 public void method1() {
  System.out.print("Person 1 ");
 }
 public void method2() {
  System.out.print("Person 2 ");
 }
}
public class Student extends Person {
 public void method1() {
  System.out.print("Student 1 ");
  super.method1();
  this.method2();
 }
 public void method2() {
  System.out.print("Student 2 ");
 }
}
public class Undergrad extends Student {
  public void method2() {
   System.out.print("Undergrad 2 ");
  }
}
```

```
Person u = new Undergrad();
u.method1();
```

Output so far:

```
Student 1 Person 1
```

type of object at runtime

Dynamic binding!

```
public class Person {
 public void method1() {
  System.out.print("Person 1 ");
 }
 public void method2() {
  System.out.print("Person 2 ");
 }
}
public class Student extends Person {
 public void method1() {
  System.out.print("Student 1 ");
  super.method1();
  this.method2();
 }
 public void method2() {
  System.out.print("Student 2 ");
 }
}
public class Undergrad extends Student {
   public void method2() {
    System.out.print("Undergrad 2 ");
   }
}
```

```
Person u = new Undergrad();
u.method1();
```

Output so far:

```
Student 1 Person 1
```

```
public class Person {
 public void method1() {
  System.out.print("Person 1 ");
 }
 public void method2() {
  System.out.print("Person 2 ");
 }
}
public class Student extends Person {
 public void method1() {
  System.out.print("Student 1 ");
  super.method1();
  this.method2();
 }
 public void method2() {
  System.out.print("Student 2 ");
 }
}
public class Undergrad extends Student {
  public void method2() {
   System.out.print("Undergrad 2 ");
  }
}
```

```
Person u = new Undergrad();
u.method1();
```

Final output:

```
Student 1 Person 1 Undergrad 2
```