Object-Oriented Programming

Overloading

Outline

- Motivation
- Parameter Signature
- Graded Exercise 3

Recall: Multiple Constructors

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

    public Person() {
        this.name = "No name";
        this.age = 0;
    }

    public Person(String name) {
        this.name = name;
        this.age = 0;
    }

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

More formally, this is called Constructor Overloading

Recall: Multiple Constructors

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

    public Person() {
        this.name = "No name";
        this.age = 0;
    }

    public Person(String name) {
        this.name = name;
        this.age = 0;
    }

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

Q: How does each constructor differ from each other?

A1: The parameters… but not exactly

A2: The signature!

Recall: Multiple Constructors

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

    ...

    public Person(String firstName) {
        this.name = firstName;
        this.age = 0;
    }

    public Person(String lastName) {
        this.name = lastName;
        this.age = 0;
    }
}
```

Q: Would this be allowed?

A: No! The signatures are the same! How would you tell the method calls apart?

```
new Person("Jose")
new Person("Rizal")
```

Hence, the signature refers to the order of parameters factoring in their data type.

Overloading

- We formally describe overloading as having multiple methods or constructors with the same name but with different signatures
  - Can be done with constructors (as we've seen already)
  - Can also be done with methods

Valid or invalid?

```
public void add(int a, int b) {
   …
}
public void add(int a, int b, int c) {
   …
}
```

Valid or invalid?

```
public void add(int a, int b) {
   …
}
public void add(int a, float b) {
   …
}
```

Valid or invalid?

```
public void add(float a, int b) {
   …
}
public void add(int a, float b) {
   …
}
```

Valid or invalid?

```
public void add(int a, int b) {
   …
}
public int add(int a, int b) {
   …
}
```

Method signature

Sequence of parameter types.

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


```
public Person(String name) { ... }
```

Method signature

Sequence of parameter types.

```
public Person(int age) { ... }
```

INT

```
public Person() { ... }
```

(empty)

Again, the same thing goes for methods!

```
public double getChange(double amt, double payment) {
    return payment - amt;
}
```


```
public double getChange(double amt, double disc, double payment) {
    return payment - amt * (100 - disc) / 100.0;
}
```

Questions? 

Exercise ☺️

Let’s head over to Canvas

Next meeting…

- [On-site] Graded Exercise 4 (*UML and coding*)

Keep learning…