Methods and Constructors

CSC 116 – Section 002

January 31, 2004

Cloning verses Copying

- Example 1:

```
//This code makes two references to one
//object
BankAccount checking = new
    BankAccount(500.00);
BankAccount savings = checking;
```

- Any change made to the savings object will affect the checking object.

Cloning verses Copying (2)

- Example 2:

  ```
  //This code makes two objects
  BankAccount checking = new
      BankAccount(500.00);
  BankAccount savings = new
      BankAccount(checking.getBalance());
  ```

- The objects checking and savings are two different objects

Null Keyword

- A reference to nothing
- The reference has no object associated with it
- Cannot run methods or access instance variables
- Ex:

  ```
  BankAccount checking = null;
  ```

This keyword

- Points to the current object
- Used in methods inside an object to reference the object, run methods of the object, or use instance variables in the object
- Don't always have to use – compiler will reference correctly for you (most of the time)
- Ex:

```
this.balance = 300.00;
```


Getter Method

- Getter Method – provides access to instance variable to retrieve (or get) value
- Ex:

```
public double getBalance() {
    return balance;
}
```

Setter Method

- Setter Method – provides access to instance variable to change (or set) value
- Ex:

```
public void setBalance(double balance) {
    this.balance = balance;
}
```


Getter and Setter Methods

- Should have a getter and setter method for each instance variable in your class.
- You should access instance variables in a class only through the getter and setter methods and not directly.

Implementing a Method

- Access Modifier – public or private
- Return Type – void, primitive, or object type
  - What the method gives back
- Method Name – lowercase letter first, than upper case letter for every other word
- Parameters – inside the parenthesis
  - Information passed to the method
  - Usually used in execution of the method

Example Method

```
public boolean withdrawMoney(double amt){
   balance = balance – amt;
   return true;
}
```

Constructor

- Initialize the object before it is first used.
- Gives all instance variables their initial value.
- Syntax:

<access modifier> <class name>([parameters]) {

    <Constructor contents>

}

Null Constructor

- Null constructor is used to initialize all instance variables to their default values.
- Null constructors take no parameters
- Ex:

```
public BankAccount() {
    this.balance = 0.0;
}
```

Complete Constructor

- Takes a parameter for each instance variable and assigns the value to the instance variable of the object
- Ex:

```
public BankAccount(double balance) {
    this.balance = balance;
}
```


Executing Constructor

- Constructor executed with new keyword
- Running constructor creates a new object of the specified type
- Ex:

```
BankAccount savings = new BankAccount(500.00);
```

Calling Methods

- Tell Java which object to call the method on
- Syntax:

  object.method();
- Ex:

  checking.deposit(500.00);

Calling Methods (2)

- Can call methods on the object that you are currently in by using the *this* keyword.
- Ex:

  this.setBalance(500.00);
- The method (A) you call must be in the same object as the method (B) you call method (A) from.
- In this case, you don't need to use the this keyword. The compiler will automatically know to use *this* as the object from which to call the method!

References

- Jason Schwarz's Lecture 6 slides: http://courses.ncsu.edu/csc116/