Method Review

GEEN163

"The general tendency is to over-design the second system using all the ideas and frills that were cautiously sidetracked on the first one."

Fred P. Brooks, Jr.

The Mythical Man-Month

Recitation Quiz

There will be a quiz in recitation on

Friday, April 15 (Today)

or

Monday, April 18

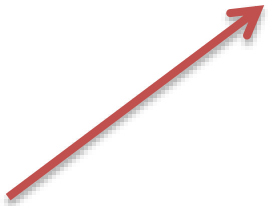
Constructor Methods

- A constructor method is automatically called when an object is created
- The name of the constructor method is always the same as the class name
- Constructor methods do not have a return type. They are not void
- Constructors may or may not have parameters

Call Constructors

- A constructor method is called when creating an object
- The method is called after the Java keyword new

```
Pet fluffy = new Pet( "Fido", "dog" );
```



- Call to constructor method

Constructors are Simple

- Most constructor methods simply copy the parameter to a class variable
- Some constructors set class variables to a constant

Bank Account Example

- Consider a class that holds information about a bank account
- A bank account object will need to contain
  - name of the account owner
  - balance of the account

Example Bank Class

```
public class BankAccount {
    private double money;       // balance in the account
    private String owner;       // name of the owner
       // constructor with two parameters
   public BankAccount(double initial, String name){
        money = initial;        // save initial balance
        owner = name;           // save name of owner
    }
      // default constructor
    public BankAccount() {
        money = 0.00;
        owner = "unknown";
    }
}
```

Multiple Constructors

- A class may have more than one constructor
- Different constructors must have a different number or type of parameters
- All constructors must have the same name as the class

- This is known as ***polymorphism***

Which is likely an accessor method?

A. void getCount ()
B. void getCount(int dog)
C. int getCount()
D. int gettysburg(int dog)

Accessor Methods

An accessor method:

- Name often begins with "get"
- Usually has no parameters
- Returns a type the same as the data value
- Is often only one line

```
public String getName() {
        return name;
  }
```

Modifier Methods

A modifier method:

- name usually starts with "set"
- usually has one parameter
- is usually void
- Often contains only one line

```
public void setName(String who) {
    name = who;
}
```

Bank Methods

- The bank account class will need methods to deposit money and withdraw money

```
public void deposit(double cash)
public void withdraw(double cash)
```

Write with your Team

- Write the two methods add or subtract from the money

```
public void deposit(double cash)

public void withdraw(double cash)
```

Possible Methods

```
public class BankAccount {
   private double money;      // balance in the account
   private String owner;      // name of the owner

 /*  method to add money to the balance */
    public void deposit(double cash) {
      money = money + cash;
   }

 /*  method to remove money from the balance */
    public void withdraw(double cash) {
      money = money - cash;
   }
}
```

Which is likely an modifier method?

A. void setCount ()
B. void setCount(int dog)
C. int setCount()
D. int settlement(int dog)

Methods with Object Parameters

Any type of data may be passed to a method

- Simple data
  - double
  - int
- Objects
- You can call methods on an object parameter or access its class instance variables

Method with an Object Parameter

```
public void transfer(
                BankAccount you,
                double      amount) {
    you.money -= amount;
    money     += amount;
}
```

Method with an Object Parameter

```
public void transfer(
                BankAccount you,
                double      amount) {
    you.withdraw(amount);
    money += amount;
}
```

Using the BankAccount class

```
BankAccount mine  = new
    BankAccount( 10000.00, "Ken");
BankAccount yours = new
      BankAccount( 10.98, "Fred");

mine.deposit(500.00);
mine.withdraw( 1.00 );
mine.transfer( yours, 5.00 );
```

javadoc

- javadoc is a tool to create webpages that document your program
- Programs have to contain specially formatted comments
- Javadoc comments start with /** and end with */
- You can put javadoc comments before methods, classes or fields

Documenting Classes

- A class may start with a Javadoc comment

```
/**
   Description of the program.
   @author your name
   GEEN163  date
*/
```

javadoc for Methods

- You should put a javadoc comment before every method defining what the method does
- The first sentence of your description will be used as the short description of the method

@param

- The @param command should be in a Javadoc comment before a method
- The @param command describes the meaning of an input parameter to the method
- For each parameter use

  @param parametername *description*

- where *description* is a sentence or more describing the use of the parameter

@return

- The @return command should be in a Javadoc comment before a method header
- You do not need @return for void methods
- @return describes the meaning of the value being returned

```
/**  Method to do something.
  * @return The calculated something or zero.
*/
```

javadoc Example

```
/**
 * This is an example.
 * @author Ken Williams
 */
public class Flag {
 /**
  * Compute the area of a rectangle.
  * @param high The height of the rectangle.
  * @param wide The width of the rectangle.
  * @return The area of the rectangle.
  */
 public double area(double high, double wide) {

                                    ...
```

Javadoc

1. is a tool for you to write web pages
2. automatically creates documentation from comments
3. Removes comments from programs
4. checks if you have correct commenting

Running Javadoc

- You can run Javadoc on one file in jGRASP from file / Generate Documentation
- On the command line, you can run javadoc by

javadoc –author –private prog1.java prog2.java

Declaring Arrays

- An array can be declared as

```
double[] dog = new double[12];
```

- This creates an array that can hold a dozen double numbers
- Note that the [brackets] on the left are empty and the [brackets] on the right contain the size

How do you make an array of BankAccount objects?

A. BankAccount[] mouse = new BankAccount(5);
B. BankAccount[] mouse = new BankAccount[5];
C. BankAccount[] mouse = new double[5];
D. BankAccount mouse = new BankAccount[5];

Array as Parameters

- An array can be passed to a method

```
boolean inThere( double[] aray, double target )
```

- When you pass an array as an argument to a method, you do not specify the size

Array Method Example

```
boolean inThere( double[] aray, double target )
        for (int i = 0; i < aray.length; i++) {
                if (aray[i] == target) {
                        return true;
                }
        }
        return false;
}
```

Recitation Quiz

There will be a quiz in recitation on

Friday, April 15 (Today)

or

Monday, April 18