Announcements

- A0 will be graded soon—everyone who submitted it gets full credit.
- It was simple enough that there is no need for us to check anything.
- A1 is due Saturday night.
- We will try to get more consultants to be available Saturday.
- Check the schedule on the course website.
- Groups: If you are going to form a group/team of two people, do it BEFORE you submit.
  - Both members must do something: one invites and the other accepts. Thereafter, only ONE member has to submit the files.
- A2: Practice with Strings
  - Assignment available now on web + CMS
  - Due on CMS by Friday, 12 September.

Homework

1. Read the text, Appendix A.1–A.3
2. Read the text, about the if-statement: A.38–A.40
3. Visit course website, click on Resources and then on Code Style Guidelines. Study
   2. Format Conventions
   4.5 About then-part and else-part of if-statement
Specifications of boolean functions

/** Return true if this Bee is male and false if not. */
public boolean isMale()

/** Return “this Bee is male”. */
public boolean isMale()

abs(-20)  Do you say, “it returns absolute value of -20” Of course not. Mathematicians say simply “that’s the absolute value of 60

/** = “this Bee is male”. */
Read as: the call isMale() equals the value of the sentence "this Bee is male".

A bit about testing

Test case: Set of input values, together with the expected output.
Develop test cases for a method from its specification --- even before you write the methods body.

Test cases for number of children

Class W (for Worker)

/** Constructor: worker with last name n, SSN s, boss b (null if none).
Precond: n not null, s in 0..999999999 with no leading zeros.*/
public W(String n, int s, W b)

/** = number of vowels in word w. 
Precondition: w contains at least one letter and nothing but letters */
public int numberOfVowels(String w) {
...
}

Developing test cases first, in “critique” mode, can prevent wasted work and errors.

How many vowels in each of these words?
creek
zyzygy

Class Object: the superest class of them all

Java: Every class that does not extend another extends class Object. That is,

public class W {...}
is equivalent to

public class W extends Object {...}

We often leave off this to reduce clutter; we know that it is effectively always there.

What is “the name of” the object?

The name of the object below is

Elephant@aa11bb24

It contains a pointer to the object – i.e. its address in memory, and you can call it a pointer if you wish. But it contains more than that.

Variable e, declared as

Elephant e;
contains not the object but the name of the object (or a pointer to the object).

Elephant@aa11bb24
Method toString

toString() in Object returns the name of the object.

Java Convention: Define toString() in any class to return a representation of an object, giving info about the values in its fields.

New definitions of toString() override the definition in Object.toString()

In appropriate places, the expression c automatically does c.toString()
c.toString() calls this method

Method toString

toString() in Object returns the name of the object.

public class W {
...}

/** Return a representation of this object */
public String toString() {
  return “Prefix: “ + isBoss() + “Suffix: “ + getName();
}

c.toString() calls this method

Another example of toString()

/** An instance represents a point (x, y) in the plane */
public class Point {
  private int x; // x-coordinate
  private int y; // y-coordinate
  ...
  /** = repr. of this point in form ”(x, y)” */
  public String toString() {
    return “(x = ” + x + “, “ + y + “);”;
  }
}

Function toString should give the values in the fields in a format that makes sense for the class.

What about this

- this keyword
  - Let’s an object instance access its own object reference
  - Example: Referencing a shadowed class field

- Intro to static components

/* = ”this object is c’s boss”.
  - Pre: c is not null. */
public boolean isBoss(W c) {
  return this == c.boss;
}

Spec: return the value of that true-false sentence.
True if this object is c’s boss, false otherwise.

- Intro to static components

/* = “b is c’s boss”.
  - Pre: b and c are not null. */
public boolean isBoss(W b, W c) {
  return b == c.getBoss();
}

Body doesn’t refer to any field or method in the object.
Why put method in object?

Intro to static components

/* = ”this object is c’s boss”.
  - Pre: c is not null. */
public boolean isBoss(W c) {
  return this == c.boss;
}
Intro to static components

```java
/** = "b is c's boss".*/
public static boolean isBoss(W b, W c)
{
    return b == c.getBoss();
}
```

Uses of static variables: Maintaining info about created objects

```java
public class W {
    private static int numObjects;
    ...
    /** Constructor. */
    public W(...) {
        numObjects++;  
    }
}
```

Example of class hierarchy: Minecraft

```java
Minecraft
```

Good example of static methods

```java
java.lang.Math
```

Java application

```java
Java application: bunch of classes with at least one that has this procedure:
```

```java
public class W {
    private static int numObjects;
    ...
    public static void main(String[] args) {
        ...
        Type String[]: array of elements of type String. We will discuss later
        
        Running the application effectively calls the method main
        Command line arguments can be entered with args
    }
}
```

Uses of static variables: Implementing the Singleton pattern

```java
public class Singleton {
    private static final Singleton INSTANCE = new Singleton();
    ...
    public static Singleton getInstance() {
        return INSTANCE;
    }
    ...
}
```

Example of class hierarchy: Minecraft

```java
Minecraft
```

Uses of static variables:

```java
Minecraft coder pack (http://mcp.ocean-labs.de)
Warning: Decompiled code with no comments ☹