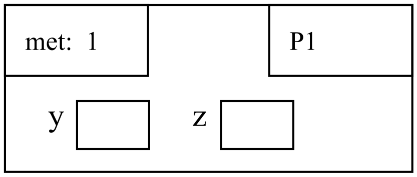
Answers to Prelim 1. CS1110, Fall 2008   David Gries djg17

1. (a) An argument is an expression that occurs in a call of a method. A precondition is a set of constraints on the parameters of a method that have to be satisfied by the arguments of a call on the method. The return type of a function is the type of value that the function returns.

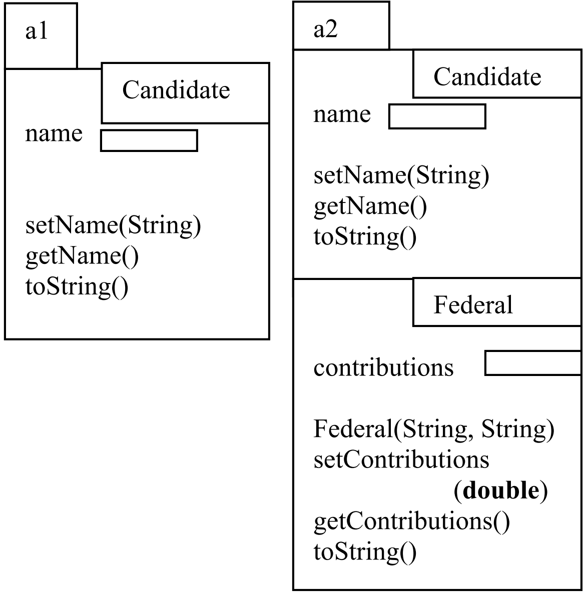
(b) b == c tests whether the values in b and c are equal, and thus whether b and c refer to the same object. b.equals(c) calls function equals in the Object partition of object b, and this has the same effect as b == c.

(c) 1: Draw a frame for the call.
2: Assign argument values to the corresponding parameters.
3: Execute the procedure body.
4: Erase the frame for the call.

(d)



2. We put the types of parameters within parentheses in showing a method; you didn't have to do this.



3.(a) 1. Create an object of class Federal. 2. Execute the constructor call Federal("Obama", 100000). 3. Yield as name of the new expression the name of the new object.
(b) toString() is overridden.
(c) 1: McCain
2. Federal: Obama
3. Obama
4. 100000.0
5. Error. No method getContributions in class Candidate.

4
```
/** An instance is for a presidential candidate */
public class Presidential extends Federal {

   private int statesWon; // Number of states this
                          //  candidate won

   /** Constructor: an instance with name n, 0 contri-
         butions, and s states won */
   public Presidential(String n, int s) {
      super(n, 0);
      statesWon= s;
   }

   /** = "this candidate won at least 25 states" */
   public boolean atLeast25() {
      return statesWon >= 25;
   }

   /** = a string representation of this instance */
   public String toString() {
      return super.toString() +
            (statesWon >= 25 ? "Good chance " : "");
   }
}
```

5
```
/** Spec given on prelim 1 */
   public static String makeID(String s) {
      int w1= s.indexOf(" "); //Index of first blank
      int w2= s.lastIndexOf(" "); // index of last blank
      String res= "" + s.charAt(0) + s.charAt(w1+1) +
         (w1 == w2 ? "" : "" + s.charAt(w2+1));
      return res.toLowerCase() + getNextID();
   }
```