Building Java Programs Chapter 5 Lecture 5-4: Assertions, do/while loops reading: 5.4 - 5.5

Logical assertions • assertion: A statement that is either true or false. Examples: · Java was created in 1995. · The sky is purple. 23 is a prime number. . 10 is greater than 20. • x divided by 2 equals 7. (depends on the value of x) An assertion might be false ("The sky is purple" above), but it is still an assertion because it is a true/false statement.

Reasoning about assertions

• Suppose you have the following code:

```
if (x > 3) {
    // Point A
x--;
} else {
    // Point B
      x++;
// Point C
// Point D
```

What do you know about x's value at the three points? Is x > 3? Always? Sometimes? Never?

Assertions in code

We can make assertions about our code and ask whether they are true at various points in the code.
 Valid answers are ALWAYS, NEVER, or SOMETIMES.

```
System.out.print("Type a nonnegative number: ");
double number = console.nextDouble();
// Point A: is number < 0.0 here? (SOMETIN
while (number < 0.0) {
       // Point B: is number < 0.0 here? (ALWA)
System.out.print("Negative; try again: ");</pre>
       number = console.nextDouble();
// Point C: is number < 0.0 here?</pre>
 // Point D: is number < 0.0 here?
                                                                             (NEVER)
```

Reasoning about assertions

• Right after a variable is initialized, its value is known:

• In general you know nothing about parameters' values: public static void mystery(int a, int b) {
// is a == 10? SOMETIMES

Assertions and loops

• At the start of a loop's body, the loop's test must be true:

```
while (y < 10) {
// is y < 10? ALWAYS
```

 After a loop, the loop's test must be false: while (y < 10) {

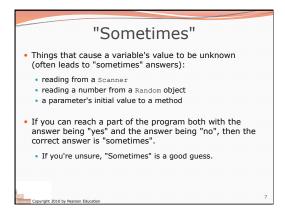
```
// is y < 10? NEVER
```

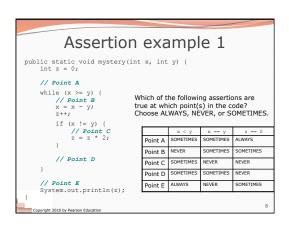
opyright 2010 by Pearson Education

Inside a loop's body, the loop's test may become false:

```
while (y < 10) {
y++;
    // is y < 10? SOMETIMES
```

bye





```
Assertion example 2
public static int mystery(Scanner console) {
  int prev = 0;
  int count = 0;
  int next = console.nextInt();
                                               Which of the following assertions are true at which point(s) in the code? Choose ALWAYS, NEVER, or SOMETIMES.
      while (next != 0) {
// Point B
           if (next == prev) {
    // Point C
                count++;
                                                                          0 prev == 0 next == prev
ES ALWAYS SOMETIMES
                                                     Point A
                                                                SOMETIMES ALWAYS
            prev = next;
next = console.nextInt();
Point B
                                                                  EVER
                                                                                            SOMETIMES
                                                     Point C
            // Point D
                                                                SOMETIMES
                                                    Point D
      // Point E
                                                    Point E
                                                                             SOMETIMES SOMETIMES
      return count;
```

```
Assertion example 3
Which of the following assertions are true at which point(s) in the code? Choose ALWAYS, NEVER, or SOMETIMES.
                                         y > 0 y % 2 ==
Point A SOMETIMES SOMETIMES
                                         Point B ALWAYS
                                                              SOMETIMES
                                         Point C ALWAYS
                                                              ALWAYS
                                         Point D ALWAYS
                                                              SOMETIMES
                                          Point E
                                                   ALWAYS
                                                              NEVER
// Point G return prod;
                                                   SOMETIMES
                                                              ALWAYS
                                          Point F
                                         Point G NEVER
```

```
while loop variations
reading: 5.4

Copyright 2010 by Paranon Education
```

```
The do/while loop

• do/while loop: Performs its test at the end of each repetition.

• Guarantees that the loop's {} body will run at least once.

do {
    statement(s);
} while (test);

// Example: prompt until correct password is typed
String phrase;
do {
    System.out.print("Type your password: ");
    phrase = console.next();
} while (!phrase.equals("abracadabra"));
```

bye 2

```
do/while question

• Modify the previous Dice program to use do/while.

2 + 4 = 6
3 + 5 = 8
5 + 6 = 11
1 + 1 = 2
4 + 3 = 7
You won after 5 tries!

• Is do/while a good fit for our past Sentinel program?
```

```
do/while answer

// Rolls two dice until a sum of 7 is reached.
import java.util.*;
public class Dice {
  public class Dice {
    public static void main(String[] args) {
        Random rand = new Random();
        int tries = 0;
        int roll2 = rand.nextInt(6) + 1;
        int roll2 = rand.nextInt(6) + 1;
        sum = roll1 + roll2;
        System.out.println(roll1 + " + " + roll2 + " = " + sum);
        tries+";
    } while (sum != 7);
    System.out.println("You won after " + tries + " tries!");
    }
}
```

```
break
• break statement: Immediately exits a loop.
• Can be used to write a loop whose test is in the middle.
• The loop's test is often changed to true ("always repeat").
while (true) {
    statement(s);
    if (test) {
        break;
    }
    statement(s);
}

**break is considered to be bad style by some programmers.
```

bye 3