Transition help for introductory code from Racket (DrScheme) and Python to Java

Racket version 5.3.6

- set DrScheme to use the Advanced Student language

  for the examples in this guide, add these helper functions to the top of your program:

```
;; displayln: ANY --> VOID
;; will act similar to System.out.println in Java
(define (displayln line)
  (begin (display line)
         (newline)))

;; in-range: NaturalNumber --> (listof NaturalNumber)
;; in-range consumes a NaturalNumber N and returns a list
;; counting up from 0 to (N-1).
(define (in-range x)
  (build-list x identity))
```

Python version 3.3.2

Java version 1.6.0_22

- *since Java requires all code to belong to a class, and include a main method as the entry point to run code, all Java code snippets (unless otherwise stated) belong within the body of the main method. For example, the actual runnable code in Java for the first and second example would look like this:*

```
public class Transition {
  public static void main(String[] args) {
    System.out.println(5 + 10);

    double LBS_TO_KILO = 0.45359237;
    System.out.println(175 * LBS_TO_KILO);
  }
}
```

  Note that you only need one main method -- you can put many of the different code snippets into the same main method and they will run in order.

Expression

Racket(DrScheme)	`(displayln (+ 5 10))`
Python	`print(5 + 10)`
Java	`System.out.println(5 + 10);`

Definition/Variable assignment

Racket(DrScheme)	`(define LBS_TO_KILO #i0.45359237)` `(displayln (* 175 LBS_TO_KILO))`
Python	`LBS_TO_KILO = 0.45359237` `print(175 * LBS_TO_KILO)`
Java	`double LBS_TO_KILO = 0.45359237;` `System.out.println(175 * LBS_TO_KILO);`

Variable re-assignment (mutation)

Racket(DrScheme)	(define age 19) (displayln age) (set! age (+ age 1)) (displayln age)		
Python	age = 19 print(age) age = age + 1 print(age)		

Java	int age = 19;
	System.out.println(age);
	age = age + 1;
	System.out.println(age);
		// ALTERNATIVE 1
		int age = 19;
		System.out.println(age);
		age += 1;
		System.out.println(age);
			// ALTERNATIVE 2
			int age = 19;
			System.out.println(age);
			age++;
			System.out.println(age);

Selection (conditional)

Racket(DrScheme)	(define JOE_AGE 25) (cond [(< JOE_AGE 18) (displayln "minor")] [(< JOE_AGE 65) (displayln "adult")] [else (displayln "senior")])

Python	JOE_AGE = 25
	if JOE_AGE < 18:
	print("minor")
	elif JOE_AGE < 65:
	print("adult")
	else:
	print("senior")

Java	int JOE_AGE = 25;
	if (JOE_AGE < 18) {
	System.out.println("minor");
	}
	else if (JOE_AGE < 65) {
	System.out.println("adult");
	}
	else {
	System.out.println("senior");
	}

Function (Java only has methods -- functions that belong to classes)

Racket(DrScheme)	(define (celsius-to-fahrenheit cdeg) (+ 32.0 (* cdeg 1.8))) (displayln (celsius-to-fahrenheit 25.0))
Python	def celsius_to_fahrenheit(cdeg): return 32 + (cdeg * 1.8) print(celsius_to_fahrenheit(25.0))

Java	public class Transition {
	public static double celsius_to_fahrenheit(double cdeg) {
	return 32.0 + (cdeg * 1.8);
	}
	public static void main(String[] args) {
	System.out.println(celsius_to_fahrenheit(25.0));
	}
	}

Repetition (definite/fixed iteration loop)

<table>
<tr><td>Racket
(DrScheme)</td><td>

```
;; ALTERNATIVE 1
;; Definite FOR loop with mutation
(define total 0)
;; for each i from 0 to 9, set total to total + i
(for-each
 (lambda (i)
  (set! total (+ total i)))
 (in-range 10))
(displayln total)
```

</td><td>

```
;; ALTERNATIVE 2
;; FOR loop as a recursive function,
;; accumulator style.
(define (sum-accum i total)
  (cond [(>= i 10) total]
        [else (sum-accum (+ i 1)
                         (+ total i))]))
(displayln (sum-accum 0 0))

;; ALTERNATIVE 3
;; FOR loop using abstraction
(displayln (foldl + 0 (in-range 10)))
```

</td></tr>
<tr><td>Python</td><td colspan="2">

```
total = 0
for x in range(10):
    total = total + x
print(total)
```

</td></tr>
<tr><td>Java</td><td colspan="2">

```
int total = 0;
for (int i = 0; i < 10; i++) {
    total += i;
}
System.out.println(total);
```

</td></tr>
</table>

Repetition (indefinite/conditional loop)

Racket(DrScheme)	```;; recursive function "WHILE" or "DO" conditional loop (define (div-seven-guess guess) (cond [(= 0 (modulo guess 7)) guess] [else (div-seven-guess (random 100))])) (displayln (div-seven-guess 1))```

Python	```import random
	guess = 1
	while guess % 7 != 0:
	 guess = random.randint(0, 99)
	print(guess)```

Java	```int guess = 1;
	while (guess % 7 != 0) {
	 guess = new java.util.Random().nextInt(100);
	}
	System.out.println(guess);```

Repetition (recursive function, performs root find midpoint algorithm on exponential functions)

Racket(DrScheme)

```
(define (root n start end base)
  (cond [(<= end start) start]
        [else
         (local [(define mid (floor (/ (+ end start) 2)))
                 (define guess (expt mid base))]
           (cond [(< guess n) (root n (+ mid 1) end base)]
                 [(> guess n) (root n start mid base)]
                 [else mid]))]))
(displayln (root 16 0 16 2))
(displayln (root 24 0 24 2))
```

Python

```
def root(n, start, end, base):
    if end <= start:
        return int(start)
    else:
        mid = int((end + start) / 2)
        guess = mid ** base
        if guess < n:
            return root(n, mid + 1, end, base)
        elif guess > n:
            return root(n, start, mid, base)
        else:
            return mid
print(root(16, 0, 16, 2))
print(root(24, 0, 24, 2))
```

Java

```
public static int root(int n, int start, int end, int base) {
    if (end <= start) {
        return start;
    }
    else {
        int mid = (end + start) / 2;
        int guess = (int) Math.pow(mid, base);
        if (guess < n) {
            return root(n, mid + 1, end, base);
        }
        else if (guess > n) {
            return root(n, start, mid, base);
        }
        else {
            return mid;
        }
    }
}

public static void main(String[] args) {
    System.out.println(root(16, 0, 16, 2));
    System.out.println(root(24, 0, 24, 2));
}
```

Object creation, field access (struct for Racket)

Racket(DrScheme)

```
(define-struct xyvector (x y))
(define (xyvector-midpoint v)
  (xyvector (/ (xyvector-x v) 2)
            (/ (xyvector-y v) 2)))
(define (xyvector-magnitude v)
  (sqrt (+ (expt (xyvector-x v) 2)
           (expt (xyvector-y v) 2))))
(define SAMPLE-XY (xyvector 10 24))
(displayln (xyvector-midpoint SAMPLE-XY))
(displayln (xyvector-magnitude SAMPLE-XY))
```

Python

```
class xyvector:
    def __init__(self, x, y):
        self.x = x
        self.y = y
    def midpoint(self):
        return xyvector(self.x / 2, self.y / 2)
    def magnitude(self):
        return (self.x ** 2 + self.y ** 2) ** .5
    def __str__(self):
        return "xyvector(" + str(self.x) + ", " + str(self.y) + ")"

SAMPLE_XY = xyvector(10.0, 24.0)
print(SAMPLE_XY.midpoint())
print(SAMPLE_XY.magnitude())
```

Java

```
// Made in a different file with name xyvector.java
public class xyvector {
    double x;
    double y;
    public xyvector(double x, double y) {
        this.x = x;
        this.y = y;
    }
    public xyvector midpoint() {
        return new xyvector(this.x / 2,
                            this.y / 2);
    }
    public double magnitude() {
        return Math.sqrt(Math.pow(this.x, 2)
                         + Math.pow(this.y, 2));
    }
    public String toString() {
        return "xyvector(" + this.x + ", "
                           + this.y + ")";
    }
}
// This should be put in Transition.java in the main method
public static void main(String[] args) {
    xyvector SAMPLE_XY = new xyvector(10.0, 24.0);
    System.out.println(SAMPLE_XY.midpoint());
    System.out.println(SAMPLE_XY.magnitude());
}
```

Test driven development

Racket(DrScheme)	(define (celsius-to-fahrenheit cdeg) (+ 32.0 (* cdeg 1.8))) (check-within (celsius-to-fahrenheit 25.0) 77.0 0.01)
Python	from cisc106 import * def celsius_to_fahrenheit(cdeg): return 32.0 + (cdeg * 1.8) assertEqual(celsius_to_fahrenheit(25.0), 77.0)

Java	import junit.framework.TestCase;
	import junit.textui.TestRunner;
	
	public class Transition extends TestCase {
	  public static double celsius_to_fahrenheit(double cdeg) {
	  return 32.0 + (cdeg * 1.8);
	  }
	
	  public static void test_celsius_to_fahrenheit() {
	  assertEquals(77.0, celsius_to_fahrenheit(25.0), 0.01);
	  }
	}
	// to run this in Eclipse, right click anywhere in the editor pane and
	// choose Run As->JUnit Test