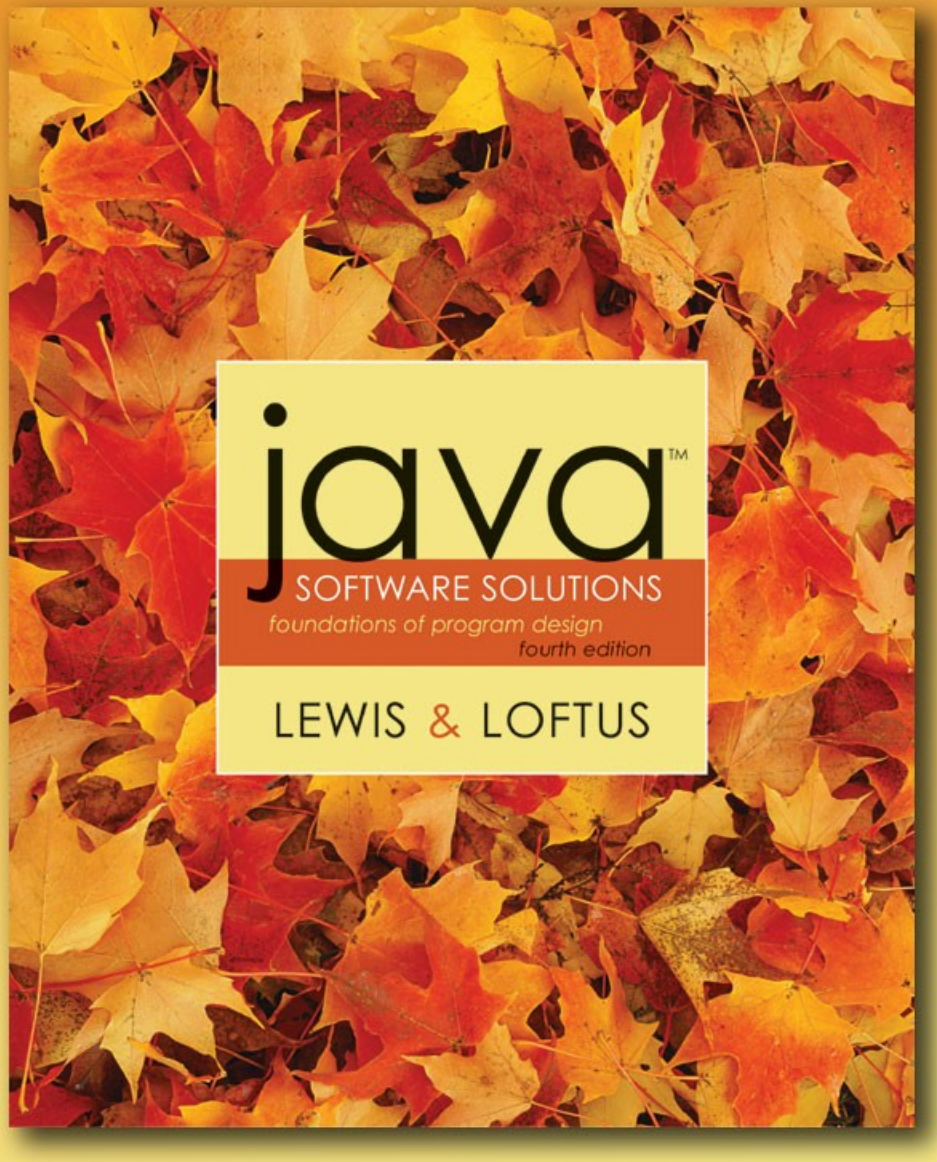
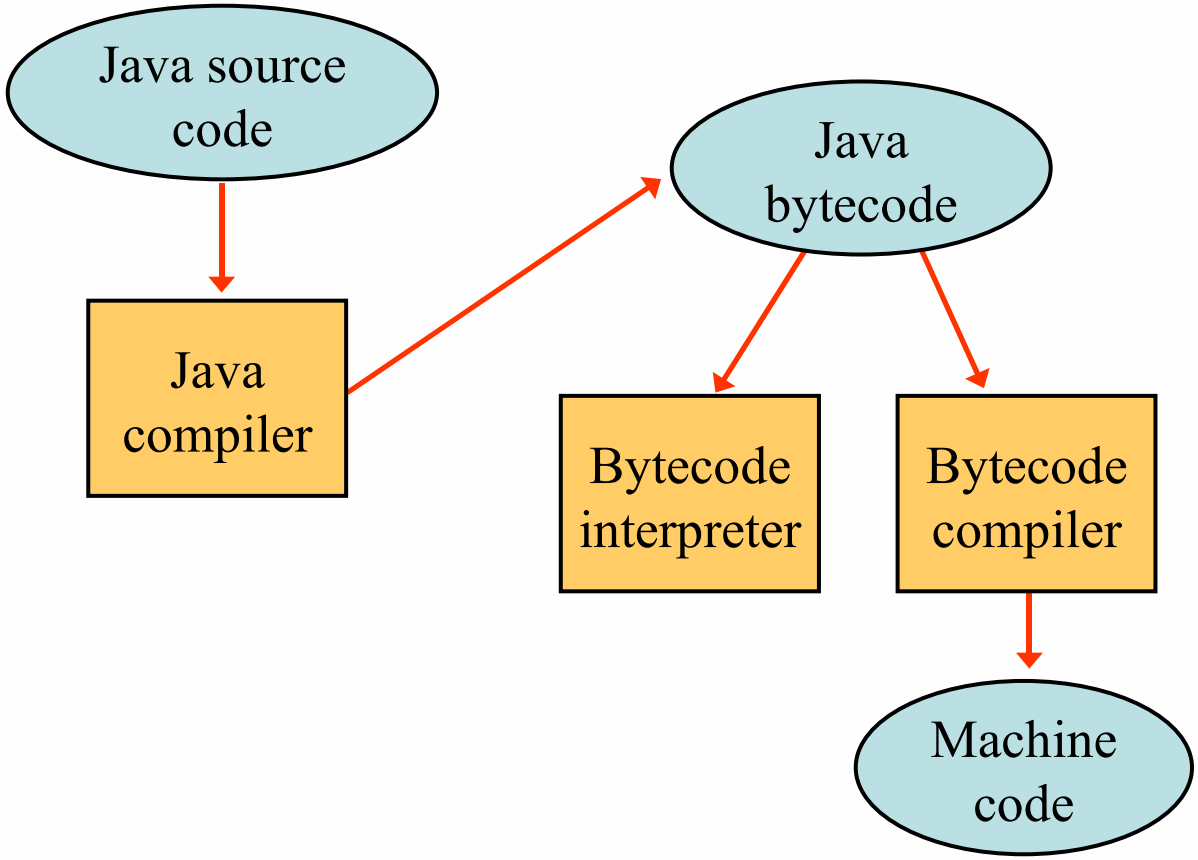
Midterm Review



Following slides are borrowed from the book's free supplement Presentation Slides, http://duke.csc.villanova.edu/jss1/

PEARSON Addison Wesley

Java Translation – Q2



the machine language for the Intel processor

Syntax and Semantics – Q3

- The ***syntax rules*** of a language define how we can put together symbols, reserved words, and identifiers to make a valid program
- The ***semantics*** of a program statement define what that statement means (its purpose or role in a program)
- A program that is syntactically correct is not necessarily logically (semantically) correct
- A program will always do what we tell it to do, not what we meant to tell it to do

Errors – Q3

- A program can have three types of errors
- The compiler will find syntax errors and other basic problems (*compile-time errors*)
  - If compile-time errors exist, an executable version of the program is not created
- A problem can occur during program execution, such as trying to divide by zero, which causes a program to terminate abnormally (*run-time errors*)
- A program may run, but produce incorrect results, perhaps using an incorrect formula (*logical errors*)

Data Conversion – Q3

- **Conversion is to convert data from one type to another. Conversions must be handled carefully to avoid losing information**
- ***Widening conversions*** **are safest because they tend to go from a small data type to a larger one (such as a `short` to an `int`)**
- ***Narrowing conversions*** **can lose information because they tend to go from a large data type to a smaller one (such as an `int` to a `short`)**
- **In Java, data conversions can occur in three ways:**
  - **assignment conversion**
  - **promotion**
  - **casting**

Assignment Conversion – Q3

- ***Assignment conversion*** **occurs when a value of one type is assigned to a variable of another**
- **If `money` is a `float` variable and `dollars` is an `int` variable, the following assignment converts the value in `dollars` to a `float`**

```
money = dollars
```

- **Only widening conversions can happen via assignment**
- **Note that the value or type of `dollars` did not change**

Data Conversion – Q3

- ***Promotion*** **happens automatically when operators in expressions convert their operands**
- **For example, if `sum` is a `float` and `count` is an `int`, the value of `count` is converted to a floating point value to perform the following calculation:**

```
result = sum / count;
```

Casting – Q3

- ***Casting*** **is the most powerful, and dangerous, technique for conversion**
- **Both widening and narrowing conversions can be accomplished by explicitly casting a value**
- **To cast, the type is put in parentheses in front of the value being converted**
- **For example, if `total` and `count` are integers, but we want a floating point result when dividing them, we can cast `total`:**

```
result = (float) total / count;
```

Question 5

- **What the result will be if x and y are integers?**

```
int x = 3;   int y = 9;

System.out.println(((y+1)/x*2+x*3/y));
```

- **3 / 2 *2.0 + 1 = ?**
- **double d = (int) 3.0/4 + 6; d=?**
  - **a) Arithmetic promotion and casting**
  - **b) Casting and assignment conversion**
  - **c) Assignment conversion and arithmetic promotion**
  - **d) Casting, arithmetic promotion ad assignment conversion**

Constants – 'final' Modifier – Q7

- A constant is an identifier that is similar to a variable except that it holds the same value during its entire existence
- As the name implies, it is constant, not variable
- The compiler will issue an error if you try to change the value of a constant
- In Java, we use the `final` modifier to declare a constant

```
final int MIN_HEIGHT = 69;
```

Constants – 'final' Modifier – Q7

- Constants are useful for three important reasons
- First, they give meaning to otherwise unclear literal values
  - For example, MAX_LOAD means more than the literal 250
- Second, they facilitate program maintenance
  - If a constant is used in multiple places, its value need only be updated in one place
- Third, they formally establish that a value should not change, avoiding inadvertent errors by other programmers

Static Modifier – Q9

- **Static methods can be invoked through the class name – no object of the class is needed, e.g.**
- **The methods of the `Math` class are *static methods* (also called *class methods*)**
- `value = Math.cos(90) + Math.sqrt(delta);`
- **Q: is it legal to call a *static method* by a object ? Why?**
- **Q: Is this illegal? Why?**
  - `Math myMath = new Math();`

Static Class Members – Q9

- **The order of the modifiers can be interchanged, but by convention visibility modifiers come first**
- **Recall that the `main` method is static – it is invoked by the Java interpreter without creating an object**
- **Static methods cannot reference instance variables because instance variables don't exist until an object exists**
- **However, a static method can reference static variables or local variables**

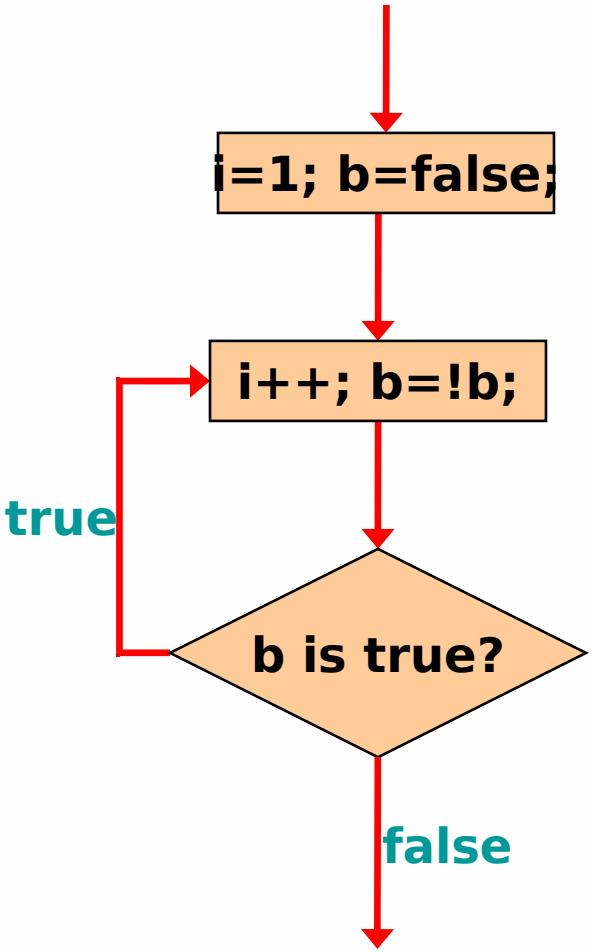
Static Variables – Q9

- **Normally, each object has its own data space, but if a variable is declared as static, only one copy of the variable exists**

```
private static float price;
```

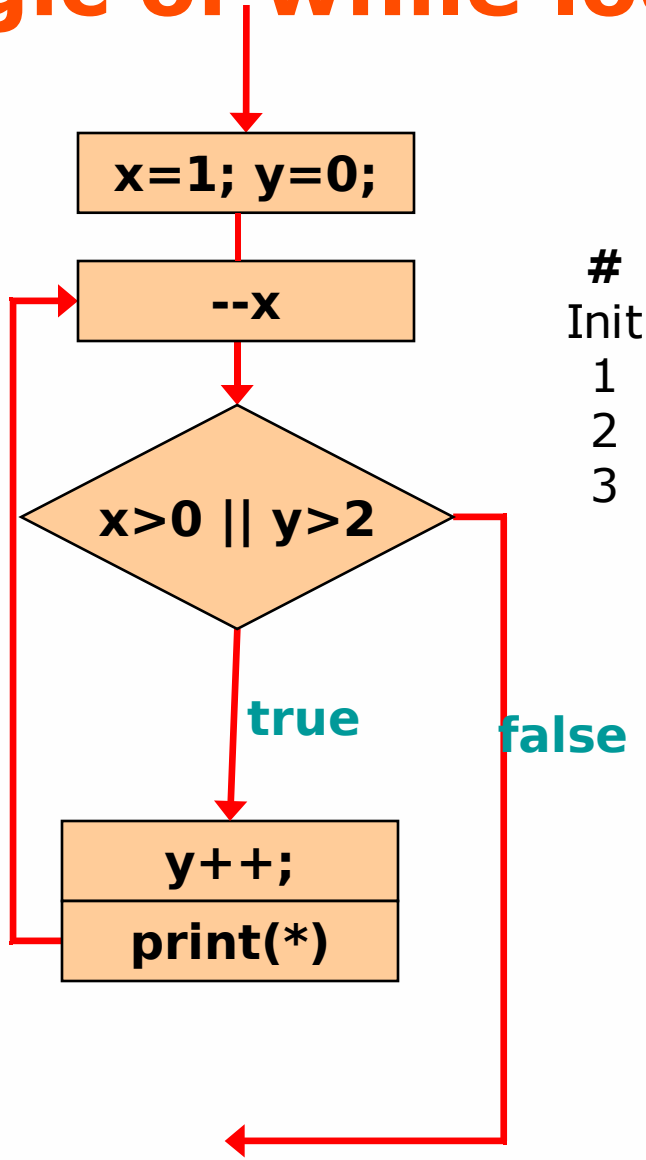
- **Memory space for a static variable is created when the class is first referenced**
- **All objects instantiated from the class share its static variables (sometimes called *class variables)***
- **Changing the value of a static variable in one object changes it for all others**

Logic of do Loop - Q-10



Iteration #	b	i	condition b
initial	false	1	--
1	true	2	true
2	false	3	false
			end of loop

Logic of while loop – Q-11



	Outside loop		*Inside loop*	
		condition		
#	**x**	**x>0 || y>2**	**y**	**print()**
Init	3		0	
1	2	true	1	*
2	1	true	2	*
3	0	false		
		end of loop		**

Let's try PART B, Q-2.