

# Become Java Certified

Ryan Allen  
LDS Church



# Personal Introduction

## Ryan Allen

### Education

- Bachelor of Computer Science, UofU, 2000
- MBA, BYU, 2007

### Work

- 1998 – IBM DB2 Internship
- 1999 – Started working with Oracle
- 2002 – Oracle Certified DBA
- 2010 – Java Certified

Started doing Java Development

# Certifications



# Class Overview

- The Certification
- Find Information
- Preparing for the Test
  - Materials
  - Total Cost
  - Hours of Study
- Scheduling the Test
- The Test
- Practice Questions
- Is it worth it?
- Questions & Answers

# The Certification

- **Was:**
  - Sun Certified Java Programmer (SCJP)
    - Exam 310-065
- **Now:**
  - [Oracle Certified Professional, Java SE 6 Programmer](#)
    - Exam 1Z0-851
    - Note: No pre-requisite on this one
  - or
  - [Oracle Certified Professional, Java SE 7 Programmer](#)
    - Exam 1Z0-804 (still in “beta” as of Mar. 2012)
    - Note: You must first become an Oracle Certified Associate, Java SE 7 Programmer (Exam 1Z0-803)

# Find Information

- <http://certification.oracle.com>

Or <http://education.oracle.com> (then click “Certification” link)

# Materials for Java 6

- Book: Sun Certified Programmer for Java 6 Exam 310-065
  - 800 pages
  - Authors: Kathy Sierra, Bert Bate
- [ePractice: Sun Certified Programmer for the Java Platform, SE 6](#)
  - 120 Questions
  - Available online for 180 days
  - Takes 3-5 days to get access once purchased

# Materials for Java 7

- OCA Book: [OCA Java SE 7 Associate Study Guide \(Oracle Press\)](#)
  - Book coming soon
  - Authors: Edward Finegan, Robert Liquori
    - I don't know how good these authors are.
- OCP Book: [OCP Java SE 7 Programmer Study Guide \(Certification Press\)](#)
  - Book coming July 22, 2012
  - Authors: Kathy Sierra, Bert Bates
- No practice tests available yet at Oracle's website.



# Cost for Java 6

- \$25 Book
  - \$65 ePractice Test
  - \$300 Exam
  - \$390 Total
- 
- Will Employer reimburse?

# Cost for Java 7

- \$35 OCA Book
- \$300 Exam
  
- \$35 OCP Book
- \$300 Exam
  
- \$670 Total
- Will Employer reimburse?

# Do Java 6 Test

- Java 7 Exams, Books, and Practice tests are not ready yet.
- I would either do Java 6 now, or wait 12 months to do Java 7.

# Time for Java 6

- 15 hrs/wk for 10 weeks = 150 hours
- Book: 80 hours (5-6 weeks)
- ePractice Test: 40 hours (3 weeks)
- Review / Memorize / Practice (1-2 weeks)

# Scheduling the Test

- [http://education.oracle.com/pls/web\\_prod-plq-dad/db\\_pages.getpage?page\\_id=51](http://education.oracle.com/pls/web_prod-plq-dad/db_pages.getpage?page_id=51)
- <http://www.pearsonvue.com/oracle>

# The Java 6 Test

- [http://education.oracle.com/pls/web\\_prod-plq-dad/db\\_pages.getpage?page\\_id=41&p\\_exam\\_id=1Z0\\_851](http://education.oracle.com/pls/web_prod-plq-dad/db_pages.getpage?page_id=41&p_exam_id=1Z0_851)
- Duration: 150 minutes
- Questions: 60
- Passing Score: 61%
- Price: \$300
  - Bring two forms of ID
    - One government issued photo id, with signature
    - One other id with signature (a credit/debit card)
- Very minor changes from original SCJP Exam
  - <https://blogs.oracle.com/certification/entry/0596>

# Practice Questions

- Practice Questions from Oracle's website
  - [http://education.oracle.com/pls/web\\_prod-plq-dad/db\\_pages.getpage?page\\_id=303&p\\_certName=SQ1Z0-851](http://education.oracle.com/pls/web_prod-plq-dad/db_pages.getpage?page_id=303&p_certName=SQ1Z0-851)
    - These questions are 10 of the 120 questions you will get if you buy the ePractice Test from Oracle for \$65.
    - They are not the real questions from the exam, but they are the exact same types of questions. (Wording, complexity, categories, “gotcha questions”, “trick questions”).
    - These prepare you for the real exam the best.

# Question 1

1) Given:

```
1. class SuperFoo {  
2.   SuperFoo doStuff(int x) {  
3.     return new SuperFoo();  
4.   }  
5. }  
6.  
7. class Foo extends SuperFoo {  
8.   // insert code here  
9. }
```

And four declarations:

```
I. Foo doStuff(int x) { return new Foo(); }  
II. Foo doStuff(int x) { return new SuperFoo(); }  
III. SuperFoo doStuff(int x) { return new Foo(); }  
IV. SuperFoo doStuff(int y) { return new SuperFoo(); }
```

Which, inserted independently at line 8, will compile?

- a) Only I.
- b) Only IV.
- c) Only I and III.
- d) Only I, II, and III.
- e) Only I, III, and IV.
- f) All four declarations will compile.



# Answer 1

**OBJECTIVE: 1.5:** Given a code example, determine if a method is correctly overriding or overloading another method, and identify legal return values (including covariant returns), for the method.

1) **Given:**

```
1. class SuperFoo {
2.     SuperFoo doStuff(int x) {
3.         return new SuperFoo();
4.     }
5. }
6.
7. class Foo extends SuperFoo {
8.     // insert code here
9. }
```

## REFERENCE:

JLS 3.0

Option E is correct. Foo doStuff() cannot return a SuperFoo and co-variant returns are legal.

**And four declarations:**

- I. `Foo doStuff(int x) { return new Foo(); }`
- II. `Foo doStuff(int x) { return new SuperFoo(); }`
- III. `SuperFoo doStuff(int x) { return new Foo(); }`
- IV. `SuperFoo doStuff(int y) { return new SuperFoo(); }`

**Which, inserted independently at line 8, will compile?**

- a) Only I.
- b) Only IV.
- c) Only I and III.
- d) Only I, II, and III.
- e) Only I, III, and IV. (\*)
- f) All four declarations will compile.

# Answer 1 Explained

- Read Chapter 2 (Sun Certified Programmer for Java 6)

- Overloading: same method name, different argument types

```
class Car {  
    Car rentCar(int x) { return new Car(); }  
    Car rentCar(int x, string y) { return new Car(); } // Overloading  
    String describeCar(){ return "I'm a car."; }  
}
```

- Overriding: same method name, same argument types (in a subclass)

```
class Ford extends Car {  
    Car rentCar(int x) { return new Ford(); } // Overriding  
    String describeCar(){ return "I'm a Ford."; }  
}
```

- Covariant Returns: As of Java 5, the declared return type can be a subtype, when overriding a method. You can only return subtypes (not super-types) of the return type.

```
class Honda extends Car {  
    Honda rentCar(int x) { return new Honda(); } // Overriding with Covariant Return  
    String describeCar(){ return "I'm a Honda."; }  
}
```

# Question 2

2) Given:

```
5. public class Buddy {  
6.     public static void main(String[] args) {  
7.         def:  
8.         for(short s = 1; s < 7; s++) {  
9.             if(s == 5) break def;  
10.            if(s == 2) continue;  
11.            System.out.print(s + ".");  
12.        }  
13.    }  
14. }
```

What is the result?

- a) 1.
- b) 1.2.
- c) 1.3.4.
- d) 1.2.3.4.
- e) 1.3.4.5.6.
- f) 1.2.3.4.5.6.
- g) Compilation fails.

# Answer 2

**OBJECTIVE: 2.2:** Develop code that implements all forms of loops and iterators, including the use of for, the enhanced for (for-each), do, while, labels, break, and continue; and explain the values taken by loop counter variables during and after loop execution.

**2) Given:**

```
5. public class Buddy {
6.     public static void main(String[] args) {
7.         def:
8.         for(short s = 1; s < 7; s++) {
9.             if(s == 5) break def;
10.            if(s == 2) continue;
11.            System.out.print(s + ".");
12.        }
13.    }
14. }
```

**What is the result?**

- a) 1.
- b) 1.2.
- c) 1.3.4. (\*)
- d) 1.2.3.4.
- e) 1.3.4.5.6.
- f) 1.2.3.4.5.6.
- g) Compilation fails.

**REFERENCE:**

JLS 3.0

Option C is correct. The continue skips the current iteration, the break ends the entire loop.

# Question 3

3) Given:

```
1. class Birds {
2.   public static void main(String [] args) {
3.     try {
4.       throw new Exception();
5.     } catch (Exception e) {
6.       try {
7.         throw new Exception();
8.       } catch (Exception e2) { System.out.print("inner "); }
9.       System.out.print("middle ");
10.    }
11.    System.out.print("outer ");
12.  }
13. }
```

**What is the result?**

- a) inner
- b) inner outer
- c) middle outer
- d) inner middle outer
- e) middle inner outer
- f) Compilation fails.
- g) An exception is thrown at runtime.

# Answer 3

**OBJECTIVE: 2.5:** Recognize the effect of an exception arising at a specified point in a code fragment. Note that the exception may be a runtime exception, a checked exception, or an error.

3) Given:

```
1. class Birds {
2.     public static void main(String [] args) {
3.         try {
4.             throw new Exception();
5.         } catch (Exception e) {
6.             try {
7.                 throw new Exception();
8.             } catch (Exception e2) { System.out.print("inner "); }
9.             System.out.print("middle ");
10.        }
11.        System.out.print("outer ");
12.    }
13. }
```

**What is the result?**

- a) inner
- b) inner outer
- c) middle outer
- d) inner middle outer (\*)
- e) middle inner outer
- f) Compilation fails.
- g) An exception is thrown at runtime.

**REFERENCE:**

JLS 3.0

Option D is correct. It is legal to nest try/catches and normal flow rules apply.

# Question 4

4) Given:

```
2. import java.io.*;
3. public class Network {
4. public static void main(String[] args) {
5. Traveler t = new Traveler();
6. t.x1 = 7; t.x2 = 7; t.x3 = 7;
7. // serialize t then deserialize t
8. System.out.println(t.x1 + " " + t.x2 + " " + t.x3);
9. }
10. }
11. class Traveler implements Serializable {
12. static int x1 = 0;
13. volatile int x2 = 0;
14. transient int x3 = 0;
15. }
```

If, on line 7, t is successfully serialized and then deserialized, what is the result?

- a) 0 0 0
- b) 0 7 0
- c) 0 7 7
- d) 7 0 0
- e) 7 7 0
- f) 7 7 7

# Answer 4

**OBJECTIVE: 3.3:** Develop code that serializes and/or de-serializes objects using the following APIs from java.io: DataInputStream, DataOutputStream, FileInputStream, FileOutputStream, ObjectInputStream, ObjectOutputStream, and Serializable.

4) **Given:**

```
2. import java.io.*;
3. public class Network {
4. public static void main(String[] args) {
5. Traveler t = new Traveler();
6. t.x1 = 7; t.x2 = 7; t.x3 = 7;
7. // serialize t then deserialize t
8. System.out.println(t.x1 + " " + t.x2 + " " + t.x3);
9. }
10. }
11. class Traveler implements Serializable {
12. static int x1 = 0;
13. volatile int x2 = 0;
14. transient int x3 = 0;
15. }
```

**REFERENCE:**

API

Option E is correct. Fields declared as transient or static are ignored by the deserialization process.

If, on line 7, t is successfully serialized and then deserialized, what is the result?

- a) 0 0 0
- b) 0 7 0
- c) 0 7 7
- d) 7 0 0
- e) 7 7 0 (\*)
- f) 7 7 7



# Answer 4 Explained

- Read Chapter 1 (Sun Certified Programmer for Java 6)
  - **Serializable**: You can save or persist the object by writing its state to an I/O stream that goes to a file or across the network to another server or client.  
**class Car implements Serializable {**  
**}**
  - **transient**: Skip / Ignore this variable when you Serialize the object. (Don't save it.)
  - **volatile**: When running multiple threads in Java, marking a variable as "volatile" will ensure that all threads will see any changes to its value.
  - **static**: Think of a "global" variable that is shared by all instances/objects of a class. This variable exists before you make any "new" objects.  
**class Car {**  
    **static int x = 5;**  
**}**  
  
...  
**System.out.println(Car.x); // This works. (No objects of Car exist yet.)**  
**Car c = new Car();**

# Question 5

5) Which regex pattern finds both 0x4A and 0X5 from within a source file?

- a) 0[xX][a-fA-F0-9]
- b) 0[xX](a-fA-F0-9)
- c) 0[xX]([a-fA-F0-9])
- d) 0[xX]([a-fA-F0-9])+
- e) 0[xX]([a-fA-F0-9])?

# Answer 5

**OBJECTIVE: 3.5:** Write code that uses standard J2SE APIs in the `java.util` and `java.util.regex` packages to format or parse strings or streams. For strings, write code that uses the `Pattern` and `Matcher` classes and the `String.split` method. Recognize and use regular expression patterns for matching (limited to: `.` (dot), `*` (star), `+` (plus), `?`, `\d`, `\s`, `\w`, `[]`, `()`) The use of `*`, `+`, and `?` will be limited to greedy quantifiers, and the parenthesis operator will only be used as a grouping mechanism, not for capturing content during matching. For streams, write code using the `Formatter` and `Scanner` classes and the `PrintWriter.format/print` methods. Recognize and use formatting parameters (limited to: `%b`, `%c`, `%d`, `%f`, `%s`) in format strings.

**5) Which regex pattern finds both 0x4A and 0X5 from within a source file?**

- a) `0[xX][a-fA-F0-9]`
- b) `0[xX](a-fA-F0-9)`
- c) `0[xX]([a-fA-F0-9])`
- d) `0[xX]([a-fA-F0-9])+ (*)`
- e) `0[xX]([a-fA-F0-9])?`

## REFERENCE:

API,

Option D is correct. The `+` quantifier finds 1 or more occurrences of hex characters after an `0x` is found.

Try <http://rubular.com/>

# Question 6

6) Given:

```
5. public class Lockdown implements Runnable {
6.     public static void main(String[] args) {
7.         new Thread(new Lockdown()).start();
8.         new Thread(new Lockdown()).start();
9.     }
10.    public void run() { locked(Thread.currentThread().getId()); }
11.    synchronized void locked(long id) {
12.        System.out.print(id + "a ");
13.        System.out.print(id + "b ");
14.    }
15. }
```

**What is true about possible sets of output from this code?**

- a) Set 6a 7a 7b 8a and set 7a 7b 8a 8b are both possible.
- b) Set 7a 7b 8a 8b and set 6a 7a 6b 7b are both possible.
- c) It could be set 7a 7b 8a 8b but set 6a 7a 6b 7b is NOT possible.
- d) It could be set 7a 8a 7b 8b but set 6a 6b 7a 7b is NOT possible.

# Answer 6

**OBJECTIVE: 4.3:** Given a scenario, write code that makes appropriate use of object locking to protect static or instance variables from concurrent access problems.

6) Given:

```
5. public class Lockdown implements Runnable {
6.     public static void main(String[] args) {
7.         new Thread(new Lockdown()).start();
8.         new Thread(new Lockdown()).start();
9.     }
10.    public void run() { locked(Thread.currentThread().getId()); }
11.    synchronized void locked(long id) {
12.        System.out.print(id + "a ");
13.        System.out.print(id + "b ");
14.    }
15. }
```

**What is true about possible sets of output from this code?**

- a) Set 6a 7a 7b 8a and set 7a 7b 8a 8b are both possible.
- b) Set 7a 7b 8a 8b and set 6a 7a 6b 7b are both possible. (\*)
- c) It could be set 7a 7b 8a 8b but set 6a 7a 6b 7b is NOT possible.
- d) It could be set 7a 8a 7b 8b but set 6a 6b 7a 7b is NOT possible.

## REFERENCE:

JLS 3.0

Option B is correct. Two different Lockdown objects are using the locked() method.

# Question 7

7) A programmer wants to develop an application in which Fizzlers are a kind of Whoosh, and Fizzlers also fulfill the contract of Oompahs. In addition, Whooshes are composed with several Wingits.

Which code represents this design?

- a) 

```
class Wingit { }
class Fizzler extends Oompah implements Whoosh { }
interface Whoosh {
    Wingits [] w;
}
class Oompah { }
```
- b) 

```
class Wingit { }
class Fizzler extends Whoosh implements Oompah { }
class Whoosh {
    Wingits [] w;
}
interface Oompah { }
```
- c) 

```
class Fizzler { }
class Wingit extends Fizzler implements Oompah { }
class Whoosh {
    Wingits [] w;
}
interface Oompah { }
```
- d) 

```
interface Wingit { }
class Fizzler extends Whoosh implements Wingit { }
class Wingit {
    Whoosh [] w;
}
class Whoosh { }
```

# Answer 7

**OBJECTIVE: 5.5:** Develop code that implements "is-a" and/or "has-a" relationships.

**7) A programmer wants to develop an application in which Fizzlers are a kind of Whoosh, and Fizzlers also fulfill the contract of Oompahs. In addition, Whooshes are composed with several Wingits.**

**Which code represents this design?**

```
a) class Wingit { }
class Fizzler extends Oompah implements Whoosh { }
interface Whoosh {
Wingits [] w;
}
class Oompah { }
b) class Wingit { }
class Fizzler extends Whoosh implements Oompah { }
class Whoosh {
Wingits [] w;
}
interface Oompah { } (*)
c) class Fizzler { }
class Wingit extends Fizzler implements Oompah { }
class Whoosh {
Wingits [] w;
}
interface Oompah { }
d) interface Wingit { }
class Fizzler extends Whoosh implements Wingit { }
class Wingit {
Whoosh [] w;
}
class Whoosh { }
```

## REFERENCE:

Sun's Java course OO226

Option B is correct. 'Kind of' translates to extends, 'contract' translates to implements, and 'composed' translates to a has-a implementation.

# Question 8

8) Given:

```
4. import java.util.*;
5. public class Quest {
6.     public static void main(String[] args) {
7.         TreeMap<String, Integer> myMap = new TreeMap<String, Integer>();
8.         myMap.put("ak", 50); myMap.put("co", 60);
9.         myMap.put("ca", 70); myMap.put("ar", 80);
10.        NavigableMap<String, Integer> myMap2 = myMap.headMap("d", true);
11.        myMap.put("fl", 90);
12.        myMap2.put("hi", 100);
13.        System.out.println(myMap.size() + " " + myMap2.size());
14.    }
15. }
```

What is the result?

- a) 4 4
- b) 5 4
- c) 5 5
- d) 6 5
- e) 6 6
- f) Compilation fails.
- g) An exception is thrown at runtime.



# Answer 8

**OBJECTIVE: 6.3:** Write code that uses the generic versions of the Collections API, in particular, the Set, List, and Map interfaces and implementation classes. Recognize the limitations of the non-generic Collections API and how to refactor code to use the generic versions. Write code that uses the NavigableSet and NavigableMap interfaces.

**8) Given:**

```
4. import java.util.*;
5. public class Quest {
6.     public static void main(String[] args) {
7.         TreeMap<String, Integer> myMap = new TreeMap<String, Integer>();
8.         myMap.put("ak", 50); myMap.put("co", 60);
9.         myMap.put("ca", 70); myMap.put("ar", 80);
10.        NavigableMap<String, Integer> myMap2 = myMap.headMap("d", true);
11.        myMap.put("fl", 90);
12.        myMap2.put("hi", 100);
13.        System.out.println(myMap.size() + " " + myMap2.size());
14.    }
15. }
```

**What is the result?**

- a) 4 4
- b) 5 4
- c) 5 5
- d) 6 5
- e) 6 6
- f) Compilation fails.
- g) An exception is thrown at runtime. (\*)

**REFERENCE:**

API

Answer: G is correct. Line 12 causes a "key out of range" exception.

# Question 9

9) Given:

```
3. import java.util.*;
4. public class ToDo {
5.     public static void main(String[] args) {
6.         String[] dogs = {"fido", "clover", "gus", "aiko"};
7.         List dogList = Arrays.asList(dogs);
8.         dogList.add("spot");
9.         dogs[0] = "fluffy";
10.        System.out.println(dogList);
11.        for(String s: dogs) System.out.print(s + " ");
12.    }
13. }
```

**What is the result?**

- a) [fluffy, clover, gus, aiko]  
fluffy, clover, gus, aiko,
- b) [fluffy, clover, gus, aiko]  
fluffy, clover, gus, aiko, spot,
- c) [fluffy, clover, gus, aiko, spot]  
fluffy, clover, gus, aiko,
- d) [fluffy, clover, gus, aiko, spot]  
fluffy, clover, gus, aiko, spot,
- e) Compilation fails.
- f) An exception is thrown at runtime.

# Answer 9

**OBJECTIVE: 6.5:** Use capabilities in the java.util package to write code to manipulate a list by sorting, performing a binary search, or converting the list to an array. Use capabilities in the java.util package to write code to manipulate an array by sorting, performing a binary search, or converting the array to a list. Use the java.util.Comparator and java.lang.Comparable interfaces to affect the sorting of lists and arrays. Furthermore, recognize the effect of the "natural ordering" of primitive wrapper classes and java.lang.String on sorting.

**9) Given:**

```
3. import java.util.*;
4. public class ToDo {
5.     public static void main(String[] args) {
6.         String[] dogs = {"fido", "clover", "gus", "aiko"};
7.         List dogList = Arrays.asList(dogs);
8.         dogList.add("spot");
9.         dogs[0] = "fluffy";
10.        System.out.println(dogList);
11.        for(String s: dogs) System.out.print(s + " ");
12.    }
13. }
```

**What is the result?**

- a) [fluffy, clover, gus, aiko]  
fluffy, clover, gus, aiko,
- b) [fluffy, clover, gus, aiko]  
fluffy, clover, gus, aiko, spot,
- c) [fluffy, clover, gus, aiko, spot]  
fluffy, clover, gus, aiko,
- d) [fluffy, clover, gus, aiko, spot]  
fluffy, clover, gus, aiko, spot,
- e) Compilation fails.
- f) An exception is thrown at runtime. (\*)

**REFERENCE:**

API

Option F is correct. The asList() method creates a fixed-size list that is backed by the array, so no additions are possible.

# Question 10

10) Given:

```
1. class x {  
2.   public static void main(String [] args) {  
3.     String p = System.getProperty("x");  
4.     if(p.equals(args[1]))  
5.       System.out.println("found");  
6.   }  
7. }
```

Which command-line invocation will produce the output found?

- a) java -Dx=y x y z
- b) java -Px=y x y z
- c) java -Dx=y x x y z
- d) java -Px=y x x y z
- e) java x x y z -Dx=y
- f) java x x y z -Px=y

# Answer 10

**OBJECTIVE: 7.2:** Given an example of a class and a command-line, determine the expected runtime behavior.

10) Given:

```
1. class x {  
2.   public static void main(String [] args) {  
3.     String p = System.getProperty("x");  
4.     if(p.equals(args[1]))  
5.       System.out.println("found");  
6.   }  
7. }
```

Which command-line invocation will produce the output found?

- a) java -Dx=y x y z
- b) java -Px=y x y z
- c) java -Dx=y x x y z (\*)
- d) java -Px=y x x y z
- e) java x x y z -Dx=y
- f) java x x y z -Px=y

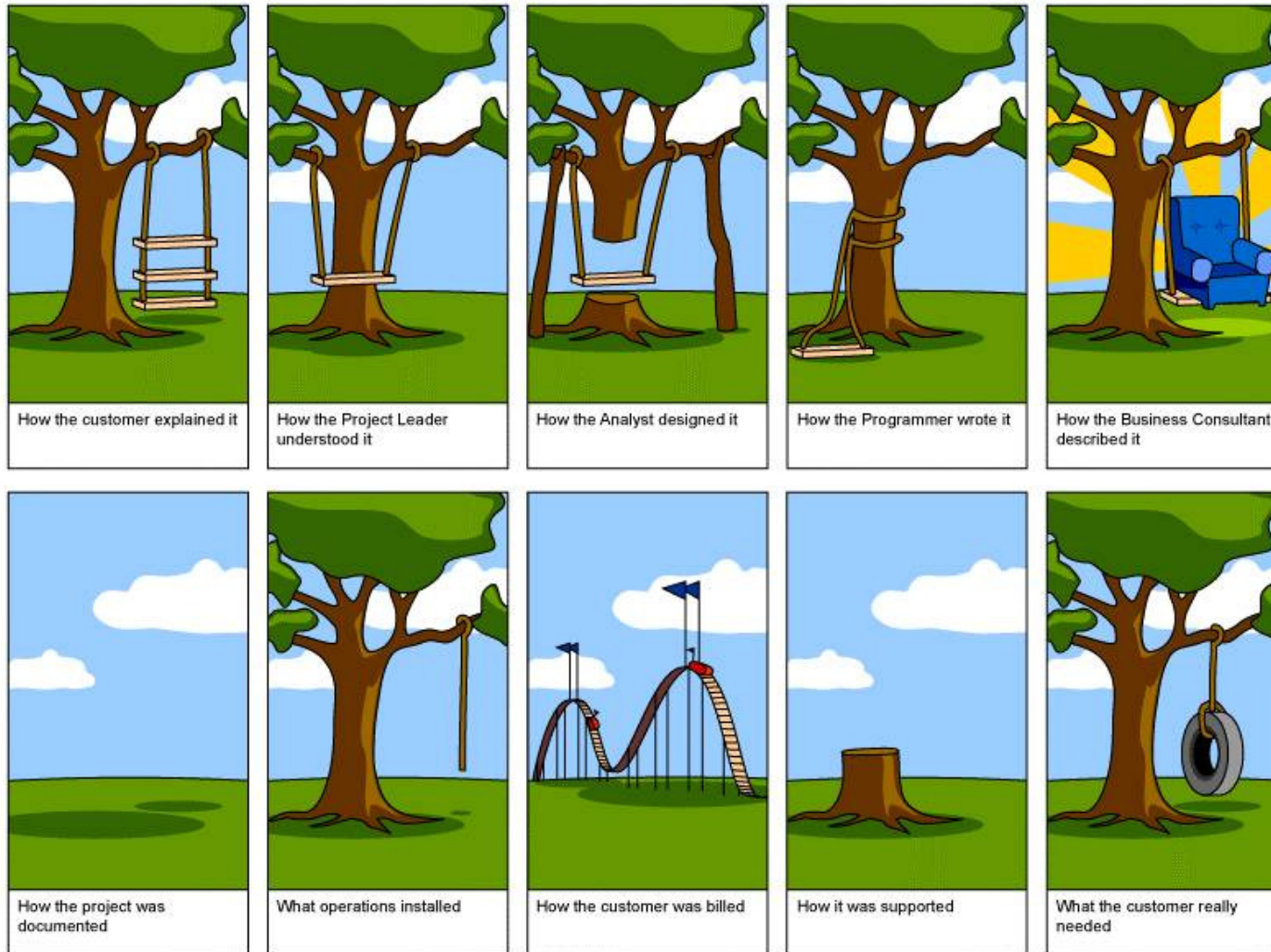
## REFERENCE:

API for java command

Option C is correct. -D sets a property and args[1] is the second argument (whose value is y)

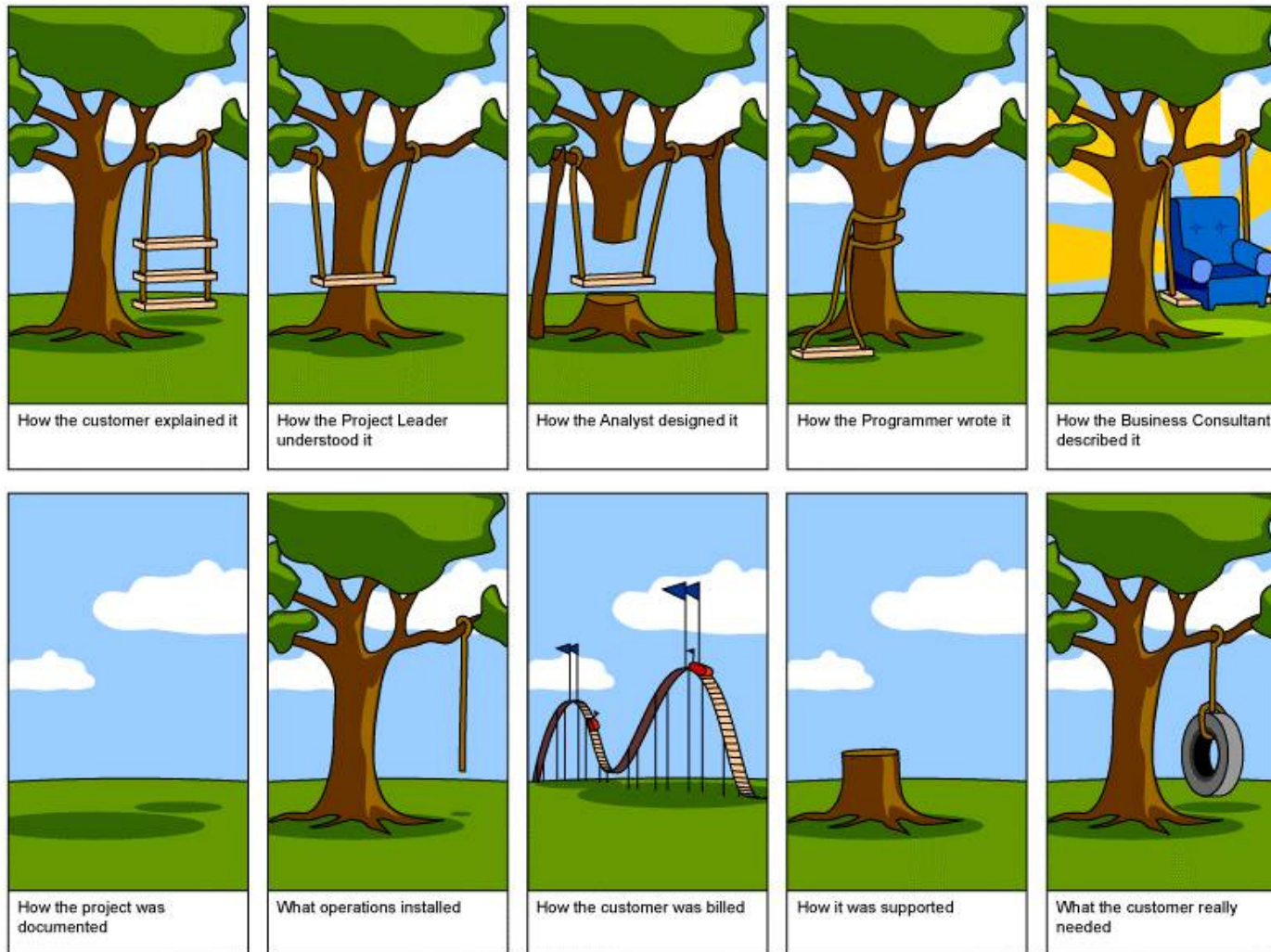
# Is the Certification worth it?

- Before getting certified, you work this way...



# Is the Certification worth it?

- After getting certified, you work this way...the same



# Is the Certification worth it?

- Java is only the beginning...
  - Web Apps
    - Spring MVC
    - Data Access (Hibernate / Spring JDBC)
    - HTML
    - CSS
    - Security (Spring Security)
    - Javascript
      - jQuery
      - Mobile (Sencha, jQuery Mobile, etc.)
      - Google Maps API
  - Web Services (SOAP & REST)



# Is the Certification worth it?

- It Depends...
  - Knowledge
  - Resume / Interviews
  - Job
  - Promotion
  - Goal for work
  - Salary
- In general...better to specialize in one thing. (Jack of all trades, master of none.)

# Questions?

# Become Java Certified

Ryan.Allen@ldschurch.org

