1) 2pts. If all constructors are named <init> how do we tell which one is supposed to be invoked in an invokestatic instruction? // 14:
invokestatic #12
Select any/all that apply.

*class name, *method signature/type, method name, method return type

2) 3pts. What is unique about constructors and how they are invoked in the Java bytecode language?
Select any/all that apply.

*They all have method name <init>
*They have void return type
*They are statically dispatched
*They are instance methods (take an object instance as first argument)
*Invoke their super class constructor first when executed
They are invoked via the new bytecode instruction
They are invoked via the invokestatic bytecode instruction
They are invoked via the invokevirtual bytecode instruction

3) 1pt. How many entries are in the local variable array for this bytecode method?
Note that this is a static method.

class Simple:
public static void main(java.lang.String[]);
Code:
0: aload_0
1: arraylength
2: istore_1
3: ldc   #2    //int 1000027
5: istore_2
6: getstatic #3 //field Simple.field1:I
9: istore_3
10: bipush 18
12: invokestatic #4 //method Simple.foo:(I)V
15: return

0, 1, 2, 3, *4, 5

4) 1pt. What type is at index 0 of this local variable array for this bytecode method?
Note that this is a static method.

class Simple:
public static void main(java.lang.String[]);
Code:
0: aload_0
1: arraylength
2: istore_1
3: ldc #2 //int 1000027
5: istore_2
6: getstatic #3 //field Simple.field1:I
9: istore_3
10: bipush 18
12: invokestatic #4 //method Simple.foo:(I)V
15: return

none, int, String ref, double, *String array ref, Simple ref

5) 1pt. How many entries are in the local variable array for this bytecode method? 
Note that this is a static method.

class Simple:
static void foo(LSimple;);
Code:
0: aload_0
1: bipush        10
3: putfield      #2 //Field Simple.ifield:I
6: aload_0
7: invokevirtual #3 //Method Simple.bar:()I
10: pop
11: return

0, *1, 2, 3, 4, 5, 6

6) 1pt. What type is at index 0 of this local variable array for this bytecode method? 
Note that this is a static method.

class Simple:
static void foo(LSimple;);
Code:
0: aload_0
1: bipush        10
3: putfield      #2 //Field Simple.ifield:I
6: aload_0
7: invokevirtual #3 //Method Simple.bar:()I
10: pop
11: return

none, int, String ref, double, *String array ref, *Simple ref

7) 1pt. How many entries are in the local variable array for this bytecode method? 
Note that this is a static method.
class Simple:
public static void main(java.lang.String[]);
Code:
    0: getstatic    #2     // Field java/lang/
        // System.err;Ljava/io/
        // PrintStream;
    3: ldc          #3     // String Hello World!
    5: invokevirtual #4     // Method java/io/
        // PrintStream.println;
        // (Ljava/lang/String;)V
    8: new          #5     // class Hello
   11: dup
   12: invokespecial #6     // Method "<init>":()V
   15: astore_1
   16: aload_1
   17: bipush        7
   19: putfield      #7     // Field field1:I
   22: aload_1
   23: invokevirtual #8   // Method getFieldVal():I
   26: istore_2
   27: return

0, 1, 2, *3, 4, 5, 6

8) 1pt. How many entries are in the local variable array for this bytecode method? Don't forget that longs and doubles take up two entries!
Note that this is an instance method.

class Simple:
public double mm(int, int, java.lang.String);
Code:
    0: iload_2
    1: i2d
    2: dstore          4 //same as dstore_4 if there was one
    4: aload_0
    5: iload_1
    6: putfield      #2     // Field ifield:I
    9: ldc2_w         #10   // double 524.0d
   12: dreturn

0, 1, 2, 3, 4, *5, *6
5 ok, but there are actually 6 b/c double takes 2 entries

9) 1pt. What type is at index 0 of this local variable array for this bytecode method? Note that this is an instance method.

class Simple:
public double mm(int, int, java.lang.String);

Code:

    0: iload_2
    1: i2d
    2: dstore 4 //same as dstore_4 if there was one
    4: aload_0
    5: iload_1
    6: putfield #2 // Field ifield:I
    9: ldc2_w #10 // double 524.0d
   12: dreturn

none, int, String ref, double, String array ref, "Simple ref"

10) 1pt. What type is at index 2 of this local variable array for this bytecode method? Note that this is an instance method.

class Simple:
    public double mm(int, int, java.lang.String);

Code:

    0: iload_2
    1: i2d
    2: dstore 4 //same as dstore_4 if there was one
    4: aload_0
    5: iload_1
    6: putfield #2 // Field ifield:I
    9: ldc2_w #10 // double 524.0d
   12: dreturn

none, "int, String ref, double, String array ref, Simple ref"

11) 2pt. Which operand stack holds the correct types for this instruction (before and after (tos:before-->tos:after) this instruction executes)?

    invokestatic #32 //method Cls.mymethod:(IILjava/lang/String;I)D

*tos:int,String,int,int-->tos:double
tos:int,int,String,int-->tos:double
tos:int,String,int,int-->tos:
tos:int,String--->tos:double
tos:-->tos:
tos:int,int,int,int-->tos:double

12) 2pt. Which operand stack holds the correct types for this instruction (before and after (tos:before-->tos:after) this instruction executes)?

    invokevirtual #21 //method Cls.myvmeth:(I)V

*tos:int,Cls--->tos:
tos:Cls,int-->tos:
tos:int-->tos:
tos:int-->tos:int
tos:-->tos:int,Cls

tos:int,Cls-->tos:int,cls

13) 1pt. Which operand stacks (more than one is possible) hold the correct types for this instruction (before and after (tos:before-->tos:after) this instruction executes)?

1: istore_2
*tos:int-->tos:
*tos:int,int-->tos:int
tos:-->tos:int
tos:long-->tos:int
tos:String ref-->tos:int

14) 1pt. Which operand stacks (more than one is possible) hold the correct types for this instruction (before and after (tos:before-->tos:after) this instruction executes)?

1: putstatic #14 //Field Cls.field1:I
*tos:int-->tos:
*tos:int,int-->tos:int
tos:-->tos:int
tos:long-->tos:int
tos:String ref-->tos:int

15) 1pt. Which operand stacks (more than one is possible) hold the correct types for this instruction (before and after (tos:before-->tos:after) this instruction executes)?

1: iload_1

tos:int-->tos:
tos:int,int-->tos:int
*tos:-->tos:int

tos:long-->tos:long,int
tos:String ref-->tos:int