

Prelim 1

CS 2110, 14 March 2017, 5:30 PM

	1	2	3	4	5	Total
Question	Name	Short answer	OO	Recursion	Loop invariants	
Max	1	36	33	15	15	100
Score						
Grader						

The exam is closed book and closed notes. Do not begin until instructed.

You have **90 minutes**. Good luck!

Write your name and Cornell **NetID**, legibly, at the top of **every** page! There are 5 questions on 7 numbered pages, front and back. Check that you have all the pages. When you hand in your exam, make sure your pages are still stapled together. If not, please use our stapler to reattach all your pages!

We have scrap paper available. If you do a lot of crossing out and rewriting, you might want to write code on scrap paper first and then copy it to the exam so that we can make sense of what you handed in.

Write your answers in the space provided. Ambiguous answers will be considered incorrect. You should be able to fit your answers easily into the space provided.

In some places, we have abbreviated or condensed code to reduce the number of pages that must be printed for the exam. In others, code has been obfuscated to make the problem more difficult. This does not mean that it's good style.

Academic Integrity Statement: I pledge that I have neither given nor received any unauthorized aid on this exam. I will not talk about the exam with anyone in this course who has not yet taken prelim 1.

(signature)

1. Name (1 point)

Write your name and NetID at the top of **every** page of this exam.

2. Short Answer (36 points.)

(a) **5 points.** Below are five expressions. To the right of each, write its value.

1. `(int)'b' == 'b'`
2. `(char>('a' +3))`
3. `new Boolean(false) == new Boolean(false)`
4. `((Object)(new Integer(3))).equals(3)`
5. `k != 0 && 5/k == 8` [note: `k` is of type `int`]

(b) **4 points.** Consider a class C with a method m . What are the two consequences of making C and m abstract?

(c) **5 points.** Function `Integer.parseInt(String s)` returns the `int` value of the integer that is in `String s`. But if s does not contain an integer, the function throws a `NumberFormatException`. Write a statement that stores in variable dn the value of the function call:

```
Integer.parseInt(somestring)
```

but stores 1 in dn if a `NumberFormatException` is thrown.

(d) **6 points.** Put a check mark before each of the following sentences that is correct and an X before each that is incorrect.

1. A class can extend only one non-abstract class but any number of abstract classes.
2. All fields in an abstract class must be public.
3. An abstract class cannot have a constructor because it cannot be instantiated.
4. If a class implements an interface, its subclasses must not implement that interface.
5. A local variable declared at the beginning of a method maintains its value from one call of the method to the next.
6. Every constructor must start with a call on a super-class constructor.

(e) **12 points.** To the right is class M1 and its subclass M2. Below is method `main` of class M1 —it belongs in class M1.

Execute a call on method `main`. Write the value that is printed by each `println` statement to the right of that `println` statement.

```
public static void main(String[] p) {
    M1 a= new M1();
    System.out.println(a.x);
    System.out.println(a.y);
    System.out.println(a.m());

    M2 b= new M2();
    System.out.println(b.x);
    System.out.println(b.y);
    System.out.println(b.m());
}
```

```
public class M1 {
    public int x= 2;
    public int y= 100;

    public M1(int x) { this.x= x; }

    public M1() { this(3); }

    public int m() {
        return this instanceof M2 ? 5 : 6;
    }
}

public class M2 extends M1 {
    public M2() { super(4); }

    public @Override int m() {
        return 100 + super.m();
    }
}
```

(f) **4 points.** What is the purpose of a constructor?

What constructor does Java insert into a class *C* if no constructor is defined in it?

3. Object-Oriented Programming (33 points)

(a) 5 points

To the right are classes K1 and K2. Method `m()` is not overridden in class K2.

Modify class K1 so that a variable will contain the number of times during execution that method `m()` is called as a method of an object of class K2 (instead of as an object of class K1 only).

Your modifications should consist of inserting a declaration in class K1 and adding code at the beginning of method `m()`.

```
public class K1 {

    public void m() {

        ...

    }

}

public class K2 extends K1 { ... }
```

(b) 10 points Below are two class declarations. Complete the bodies of the constructor and function `toString` in class `Surgeon`. Be careful; pay attention to access modifiers.

```
public class Doctor {
    private String name;

    /** A doctor named n.
     * Precond.: no space in n */
    public Doctor(String n) {
        name= n;
    }

    /** Return this doctor's name */
    public String toString() {
        return name;
    }
}
```

```
public class Surgeon extends Doctor {
    private int ops; //no. of ops performed
    /** Constructor: instance with name n
     *                and op operations
     * Precond.: no space in n */
    public Surgeon(String n, int op) {

    }

    /** Return this surgeon's name, a
     * space, and number of operations. */
    public String toString() {

    }

}
}
```

(c) 5 points Complete the body of method *equals*, which belongs in class *Doctor*:

```
/** Return true iff ob is a Doctor and
 * ob has the same name as this Doctor. */
public @Override boolean equals(Object ob) {

}
}
```

(d) 5 points Write down the steps in evaluating a new-expression `new C(args)` .

(e) 8 points

Consider the interface and class declarations given below. Next to each piece of Java code in the righthand column, write whether it produces no error, a run-time error, or a compile-time error. (Assume that each piece is independent of the others.)

Hint: It will help to draw objects of the classes.

```
interface I1 {...}
interface I2 {...}
interface I3 extends I1 {...}
class C1 implements I1 {...}
class C2 implements I2 {...}
class C3 implements I3 {...}
class C4 extends C3
    implements I2 {...}
```

- (a) `I2 a= new I2();`
- (b) `I2 b= new C2();`
- (c) `C3 c= new C4();`
- (d) `C2 d= new C4();`
- (e) `C4 e= new C3();`
- (f) `C4 f= (C4)(new C3());`
- (g) `I1 g1= new C1();`
`C4 g2= new C4();`
`g1= g2;`
- (h) `I1 g1= new C4();`
`I2 g2= new C2();`
`g2= g1;`

4. Recursion (15 Points)

(a) Write the body of recursive function nf , whose specification and header appear below. Do not use loops. Use only recursion. Here is a restriction, which should help you hone in on a simple solution: The only *String* functions you should use are *charAt*, *length*, and *substring*.

```
/** Return the number of times the first
 * char of s appears in a row at the beginning of s.
 * Precondition: s is not null and contains at least 1 char.
 * Example: nf("bbbcb$b") = 3.
 * Example: nf("bcb$bbb") = 1. */
public static int nf(String s) {

}
}
```

(b) Below is function `comfy`. It is complete except for the base-case if-condition. Circle all possible expressions from the list below that could be used for the base-case if-condition.

1. `s.length() < 3`
2. `s.length() ≤ 3`
3. `s.length() == 3`
4. `Integer.parseInt(s) < 1000`
5. `s.length() == 0`

```
/** Return s formatted by adding a comma before every third digit.
 * E.g. 1000 is formatted as 1,000, 56 is 56, and 1234567 is 1,234,567.
 * Precondition s is a non-signed integer and the leftmost digit is not 0. */
public String comfy(String s) {
    if ( base-case if-condition ) return s;
    return comfy(s.substring(0,s.length()-3)) + ',' + s.substring(s.length()-3);
}
```

5. Loop Invariants (15 points)

(a) **2 points** State the formula for the number of values in array segment $b[h..k - 1]$.

(b) **13 points** Consider the following precondition, invariant, and postcondition. The postcondition has two alternatives —either section $b[h..j - 1]$ or section $b[j + 1..k]$ is empty (the other one might be, but it is not necessary).

Precondition:	b	0		j	x		n
			?			?	

Invariant:	b	0		h		j	x		k		n
			$\leq x$?			?		$\geq x$	

Postcondition:	b	0		j	x		k		n
			$\leq x$?		$\geq x$	

OR	b	0		h		j	x		n
			$\leq x$?			$\geq x$	

Write a loop with initialization that uses the invariant given above to implement the comment given below. Thus, the loop should continue as long as both ? sections are non-empty. Assume that b , j , and n are already initialized. Identifier x can't be used in the program; it just stands for the value in $b[j]$. Don't declare variables, but do assign appropriate values to h and k wherever necessary. To swap $b[i]$ and $b[j]$, just say, "Swap $b[i]$ and $b[j]$." Your grade depends only on how well you use the four loopy questions to write the code.

```
// Given the Precondition as shown above, swap values of array
// segment b[0..n] so that the Postcondition holds.
```