

# Computer Science 145

Final Exam—Fall 2011

Name: \_\_\_\_\_

Problem	Score	Possible
1		6
2		8
3		6
4		6
5		6
6		6
7		6
8		6
Total		50

This is a closed-book, no-calculator, no-electronic-devices, individual-effort exam. You may reference one page of handwritten notes. All answers should be clearly written. Questions that require code should be written using correct Java syntax. Please do all your work on these pages. Partial credit will be given if work is shown and is partially correct. You may write `SOP` to represent `System.out.println`.

Class	Method/Constructor	Description
Scanner	<code>Scanner(System.in)</code> <code>Scanner(String text)</code> <code>String next()</code> <code>double nextDouble()</code> <code>boolean nextBoolean()</code> <code>int nextInt()</code>	create <code>Scanner</code> for parsing <code>System.in</code> create <code>Scanner</code> for parsing <code>text</code> get next delimited word get next delimited double get next delimited boolean get next delimited integer
String	<code>int length()</code> <code>char charAt(int i)</code> <code>boolean endsWith(String other)</code> <code>boolean startsWith(String other)</code>	get number of characters get the character at index <code>i</code> return true if this <code>String</code> ends with <code>other</code> return true if this <code>String</code> starts with <code>other</code>
Random	<code>Random()</code> <code>nextInt(int i)</code>  <code>nextDouble()</code>	create a random number generator. get a random number between 0 and <code>i - 1</code> , inclusive.  get a random number between 0.0 and 1.0.

1. *Feliz Naverdad*

Fill in the blanks below so that each code snippet prints the boolean value `true` to `System.out`.

(a) `boolean b = _____;`  
`System.out.println(!b);`

(b) `String s = _____;`  
`System.out.println(s.length() == 2);`

(c) `System.out.println(5 != _____);`

(d) `System.out.println(false || _____);`

(e) `Scanner in = new Scanner("_____");`  
`System.out.println(in.hasNextDouble());`

(f) `Random gen = new Random();`  
`System.out.println(gen.nextInt(_____) < 100);`

## 2. Us && Vegas || Bust

Suppose  $a$ ,  $b$ , and  $c$  are boolean variables. For each row in the table below, suppose variables  $a$ ,  $b$ , and  $c$  had the values listed in the row and evaluate the expression in the rightmost column accordingly. Write T if the expression evaluates to **true**; write F otherwise. You may optionally use the complete blank column to evaluate subexpressions.

a	b	c		$!a \ \&\& \ !b \    \ c$
T	T	T		
F	T	T		
T	F	T		
F	F	T		
T	T	F		
F	T	F		
T	F	F		
F	F	F		

3. *Less Work, More Play*

Write a `static` method named `foo` that takes a `String` argument and an `int` argument `n`. It prints the `String` argument `n` times.



5. *Object-Oriented*

Write a class `Circle` having:

- (a) a `public` constructor taking a `double` argument for the circle's radius
- (b) a `public` method `getArea` that returns the circle's area ( $\pi \times radius^2$ ) as a `double`

Any instance variables must be `private`. Use `Math.PI` for the value of  $\pi$ .

## 6. *Classification*

You've downloaded the source code for some address book software, and it's evident that the original programmer knew little about organizing data into classes. You find the following code in main:

```
String[] names = new String[peopleCount];
Date[] birthdays = new Date[peopleCount];
String[] emails = new String[peopleCount];

for (int i = 0; i < peopleCount; ++i) {
    names[i] = getNextName();
    birthdays[i] = getNextBirthday();
    emails[i] = getNextEmail();
}
```

Instead of spreading each person's information across three arrays, write a class `Person` that holds all three pieces of a `Person`'s state. Include a constructor, but omit any other methods.

7. Rewrite the provided array-filling code on the previous page using one `Person` array—and no other arrays. You can assume `peopleCount` has been defined and the helper methods like `getNextName` have been implemented and are valid.

8. *Fadeout*

Suppose you have an grayscale image stored in a 2-D `int` array named `pixels`, which has already been declared and allocated. Write code to reduce each pixel in the array to half its brightness. Truncate any fractional remainders.