

CSCI 261J
1st Midterm Exam
February 15, 2013

Name: _____

| | | | | |
|-----------|----|----|----|-------|
| Question: | 1 | 2 | 3 | Total |
| Points: | 20 | 15 | 15 | 50 |
| Score: | | | | |

Question 1 (20 points)

- (a) Specify (by writing either **true** or **false** alongside each statement) the values of the six **boolean** variables computed as follows:

```
double x = 3.14;
int y = 3;
boolean b1 = x/4>0;
boolean b2 = y/4>0;
boolean b3 = x>y && y<0;
boolean b4 = x<y && y>0;
boolean b5 = x>y || y<0;
boolean b6 = x<y || y>0;
```

- (b) What is printed by the following program fragment?

```
int j = 1;
int k = 11;
for (int i=1; i<10; i+=2) {
    j = j*2;
    k = k%2;
    System.out.println("i="+i+" j="+j+" k="+k);
}
```

- (c) Assume that you have stored the radius of a circle in a **float** **r**. Using a standard Java class to obtain an approximation to π (a **double**), write a single Java statement that computes the area of the circle and stores it in a **float** **a**.

Question 2 (15 points)

- (a) Draws a target with five rings with diameters 10, 20, 30, 40 and 50 pixels, all centered at coordinates $(x,y) = (100,100)$ pixels. The alternating ring colors are black, white, black, white, and black.

```
public static void drawTarget(Graphics g) {
```

```
}
```

- (b) Prints the sum of only the odd integers in the specified set $\{a, b, c, d\}$. If all four of the integers are even, prints zero.

```
public static void printSumOfOdd(int a, int b, int c, int d) {
```

```
}
```

- (c) Returns the sum of the cubes of the first n whole numbers, that is, $1 + 8 + 27 + \dots + n^3$. (Use simple multiplication to compute the cube of each of the integers.)

```
public static int sumOfCubes(int n) {
```

```
}
```

Question 3..... (15 points)

Assume that you have a class `Stopwatch` that works much like a real stopwatch, with the following methods

- `public void start()` starts the stopwatch
 - `public void stop()` stops the stopwatch
 - `public double time()` returns (does not print!) the number of seconds that have elapsed between start and stop
 - `public void reset()` resets the stopwatch so that the number of elapsed seconds is zero
- (a) Write a Java program fragment (not a complete program) that constructs a `Stopwatch` and then uses its methods defined above to determine and print the number of seconds required to compute `sumOfCubes(1000000)`. (Here, do not write any code for the class `Stopwatch`; see part (b) below.)

- (b) Assume that the method `Clock.seconds()` returns a `double` equal to the number of seconds that have elapsed since January 1, 2013. Use this method to write the Java class `Stopwatch` that implements the four methods specified above. Include in your class everything necessary to implement these methods.