

This exam is: **closed-book**, **NO electronic devices allowed**, and **closed-notes**. The exception is the “sage page” of the designated size on which you may have notes to consult during the exam.

Be sure you: **Provide legible answers in designated areas (credit will not be given for work that is difficult to read or not where expected)**, **Ensure you clearly fill in a single circle on multiple choice questions**, **Use indentation of your code to show its structure (but don’t dwell on exact punctuation/syntax)**, **Leave the exam stapled together in its original order**, **Do *NOT* attach any other pages to the exam**. You are welcome to use the blank space on the exam for any scratch work.

If you need to leave the room for any reason prior to turning in your exam, you must leave your exam and any electronic devices with a proctor.

You must complete all the identifying information below correctly. Failure to do so is grounds for a zero on this exam:

1. Name (**print clearly**): _____

2. Student ID (**print clearly; 1 digit per underline**): _____

3. Which time do you typically attend studio (fill one)

- 11:30—1:00 1:00—2:30 2:30—4:00 4:00—5:30

4. Which Urbauer lab do you typically sit in (best guess; fill one)

- 214 216 218 222

5. You must sign the pledge below for your exam to count. The penalty for cheating will be decided during academic integrity review, but the instructors will recommend an F in this course as the minimum penalty.

I have read the instructions on this page and I will neither give nor receive any unauthorized aid on this exam.

(Sign above)

⇒ Do not proceed until told to do so! ⇐

⇒ Initial the top right corner of each page before starting ⇐

1. (18 points) For each expression below indicate the type and value of the result. Use quotation marks to indicate strings. The first row has been completed as an example.

Expression	Result Type	Result Value
"Good Luck!"	<input type="radio"/> boolean <input type="radio"/> double <input type="radio"/> int <input checked="" type="radio"/> String	Good Luck!
$(3/2) \leq 1$	<input type="radio"/> boolean <input type="radio"/> double <input type="radio"/> int <input type="radio"/> String	
$4 * 5 / 3$	<input type="radio"/> boolean <input type="radio"/> double <input type="radio"/> int <input type="radio"/> String	
$1 + 2 + "4"$	<input type="radio"/> boolean <input type="radio"/> double <input type="radio"/> int <input type="radio"/> String	
$1.0 / 2$	<input type="radio"/> boolean <input type="radio"/> double <input type="radio"/> int <input type="radio"/> String	
$"1" + 2 + 4$	<input type="radio"/> boolean <input type="radio"/> double <input type="radio"/> int <input type="radio"/> String	
$\text{true} \ \&\& \ (4 < 3)$	<input type="radio"/> boolean <input type="radio"/> double <input type="radio"/> int <input type="radio"/> String	
$(! \text{true}) \ \ \text{false}$	<input type="radio"/> boolean <input type="radio"/> double <input type="radio"/> int <input type="radio"/> String	
$15 \% 4 + 1.0$	<input type="radio"/> boolean <input type="radio"/> double <input type="radio"/> int <input type="radio"/> String	
$(\text{double}) 2 * (1/2) + 1$	<input type="radio"/> boolean <input type="radio"/> double <input type="radio"/> int <input type="radio"/> String	

2. (4 points) Indicate the most appropriate data type for each concept:

(1) Your WashU GPA:

- boolean double int String

(2) An indicator that a person is a licensed driver:

- boolean double int String

(3) The number of tires on a vehicle:

- boolean double int String

(4) A country's capital city:

- boolean double int String

3. (4 points) Given the following Java statement:

```
a = 3+2*Math.sqrt(b);
```

(1) Which operation is completed first?

- = + Math.sqrt() *

(2) Which operation is completed second?

- = + Math.sqrt() *

(3) Which operation is completed third?

- = + Math.sqrt() *

(4) Which operation is completed last?

- = + Math.sqrt() *

4. (4 points) In each of the following snippets of code circle the code corresponding to each term:

(1) The *predicate*

```
for(int i=0; i<12; i++) {  
    System.out.println(i);  
}
```

(2) The *loop body*

```
for(int i=0; i<12; i++) {  
    System.out.println(i);  
}
```

(3) The *boolean expression*

```
for(int i=0; i<12; i++) {  
    System.out.println(i);  
}
```

(4) The *variable declaration*

```
for(int i=0; i<12; i++) {  
    System.out.println(i);  
}
```

5. (7 points) True or False and multiple choice. Fill in the correct circle:

(1) The statement `--i;` has the same effect on `i` as `i=i-1;`

- True False

(2) Doing `2/0.0` results in:

- 0 `Double.MAX_VALUE`
 Infinity NaN

(3) The range of Java's `int` is closest to:

- 2,000 to 2,000 0 to 2,000,000
 -2,000,000 to 2,000,000 -2,000,000,000 to 2,000,000,000

(4) Including one loop within another loop is an example of:

- Hyperlooping Loopification
 Nesting Reiterating

(5) Given the following Java statement:

```
int [][] array = new int [10][5];
```

`array.length` will be:

- 4 5 6
 9 10 11
 36 50 66

(6) The underlined portion of the following:

```
double d = 0;
```

```
...
```

```
System.out.println("There are " + (int)d + " items left");
```

is an example of:

- explicit casting intification
 refactoring type widening

(7) In Java, operations like: `"Hello" + "World"` are called:

- String Concatenation String Summation
 String Theory String Nesting

6. Show the output of each snippet of code in the area provided. Be as accurate as possible, including spacing and line usage. All output should start on the first dotted line on the right side of the paper (the leftmost dot corresponds to the left side of the console window). *The first line of the first problem has been filled in as an example.*

(1) (1 point)

```

for (int i=0;i<3;i++) {
    if (i%2==0) {
        System.out.println(i);
    }
}

```

0
.....
.....
.....
.....
.....

(2) (4 points)

```

for (int i=0;i<3;i++) {
    for (int j=10; j>8; j--) {
        System.out.println("i:"+i+" j:"+j);
    }
}

```

.....
.....
.....
.....
.....
.....
.....
.....

(3) (4 points)

```

int i=11;
int j=14;
while(i>1 || j<19) {
    System.out.println(i + " " + j);
    i = (i+1)%4;
    j = j+2;
}

```

.....
.....
.....
.....
.....
.....
.....
.....

(4) (7 points)

```
for(int i = 10; i>0; i--) {  
    if(i%4==0 || i>7) {  
        System.out.print(i + ":" + i%3);  
        if(i>4) {  
            System.out.println(" A");  
        } else {  
            System.out.println(" B");  
        }  
    }  
}
```

.....
.....
.....
.....
.....

(5) (4 points)

```
int [] data={ 1, 2, 3, 4, 5};  
data[1] = data[1]+data[2];  
data[4]--;  
for(int i: data) {  
    System.out.println(i);  
}
```

.....
.....
.....
.....
.....
.....

(6) (4 points)

```
String [] vals={"i","j","k","l","m","n"};  
int [] next={ 5, -1, -1, 0, -1, 2};  
int current = 3;  
while(current >= 0) {  
    System.out.println(current+": "+vals[current]);  
    current = next[current];  
}
```

.....
.....
.....
.....
.....
.....

8. (13 points) Square² Inc. would like to help students learn perfect integer squares (i.e., numbers that are the square of an integer). Write a program that will: 1) pick a random value between 1 and 1,000,000 (inclusive), then 2) identify the first perfect square that is *larger* than the randomly selected value, 3) repeatedly prompt the user to enter a value until they enter the perfect square, and 4) print a message indicating when they have entered it correctly, what the square is, and what its square root is.

An *example* should behave like the following (try to match this format exactly):

Shown in an Args Processor Window	User Enters
Enter the first square greater than 39996	39997
Enter the first square greater than 39996	39999
Enter the first square greater than 39996	40000

At the end, when the correct value is guessed, this run would print:

“You got it, 40000 is 200 squared”

Show work on next page

Show work on next page

Show work on next page

Show work on next page

Show work on next page

Show work on next page

Show work on next page

Show work on next page

