This is Exam #1 in CSMC B113 from Fall 2018.

Students had 80 minutes to complete the exam. Note that this was a closed-book, closed-notes exam.

This is being provided purely as an example of a past exam, and is not necessarily indicative of the length, difficulty, format, or set of topics covered on the exam that will be given in Fall 2020.
Question 1 (15 points)

Part 1. Basic Java Knowledge. Write Java command(s) to do the following:

(1) Define a floating point variable named, \( x \) ___________________________________

(2) Set the variable \( x \) (from above) to 42.3 ___________________________________

(3) Increment the value in \( x \) by 5.9 ___________________________________

(4) Compute the square root of \( x \) and place it in \( x \) ___________________________________

(5) Convert the value of \( x \) to an integer and save the result in an integer variable \( y \): ___________________________________

(6) Convert \( \theta \) (type double) from degrees to radians ___________________________________

(7) Define an array named \( a \) of type double. ___________________________________

(8) Create the array \( a \) from (7) above to have 100 elements.
_________________________________________________________________________

(9) Set the element at index 42 in \( a \) (above) to the value of \( \pi \). _______________________

(10) Set the element at index 39 in \( a \) (above) to a random number between [0..1) _______________________

Note that there were also five additional questions related to bash commands; those have been removed from this document since bash commands will not be on the Fall 2020 exam.
Question 2 (10 points) Write Java commands to exchange the contents of two integer variables \( x \) and \( y \).

\[
\text{int } x = \ldots, \text{ int } y = \ldots;
\]

Question 3 (10 points) Write Java commands to create an array named \( x \) of \( N \) elements and fill it up with random integers between \([10..10000]\) (inclusive).

Question 4 (10 points) \( x \) is the same array as in Question 3 above. Study carefully the Java commands shown below:

\[
\text{int } r = x[0];
\text{for (int } i=1; i < x.\text{length}; i++) \{ \\
\text{ if (} x[i] < r \text{) } \{ \\
\text{ r = } x[i]; \\
\text{ } \}
\}
\]

(a) How many times will the for-loop be executed? _____________

(b) Describe, in one sentence, what the set of commands is computing.
Question 5 (10 points) Given three integer variables, \( x, y, z \) (assume already defined) write Java commands to assign to a variable \( \text{max} \) (you have to define it) the largest value in \( x, y, \) and \( z \).

Question 6 (10 points) Here is an algorithm for computing the GCD of two numbers \( a \) and \( b \):

while \( a \neq b \)
  if \( a > b \)
    \( a \leftarrow a - b \)
  else
    \( b \leftarrow b - a \)

Assuming \( a \), and \( b \) are already defined (as integers), write Java commands to code the above algorithm.
Question 7 (10 points)
What will be the exact output when the following Java statements are executed:

```java
int n = 5;

for (int i=0; i < n; i++) {
    for (int j=i+1; j < n; j++) {
        System.out.println(i + " " + j + " " + (i+j));
    }
}
```
Question 8 (10 points)
Write a complete Java program that displays all odd powers of 3 between 0 and 20 (i.e. $3^i$ where $0 < i \leq 20$ and $i$ is odd). Output a table (see box) showing the power 3 is being raised to, as well as the result, on each line. You may use a TAB character (‘\t’) to separate and align the two numbers on each line.

Example Output

<table>
<thead>
<tr>
<th>i</th>
<th>$3^i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>27</td>
</tr>
<tr>
<td>5</td>
<td>243</td>
</tr>
<tr>
<td>7</td>
<td>2187</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Question 9 (15 points) Write a **complete Java program** called **LuckySeven** that simulates the tossing of two six-sided dice and computes the estimated probability of obtaining a sum of 7 on each roll. Your program should input N, the number of trials as an integer from the command line. Here is a sample output:

```
$ java LuckySeven 10000
The probability of obtaining a 7 in 10000 trials is 17%
```