

Computer Sciences 302
Exam 1 Information & Sample Questions

Below you'll find information about the first midterm exam and sample exam questions. This sampling is intended to show you a variety of multiple-choice question styles. It does not represent the length or difficulty of the actual exam. The exam will have more questions and cover a broader range of topics than what is presented here. For a list of topics for exam 1, see the course web pages. Note the following logistics about the exam:

- This semester the first midterm exam will be composed of 25 multiple-choice questions, each worth 3 points, with a maximum score of 72 points. Note that you do not need to correctly answer all of the questions to achieve 100% on the exam. We're providing one bonus question, but note that it will not count as extra credit if you score higher than 100%.
- The exam is designed to be 90 minutes in length, but we'll allow an additional 30 minutes for a total exam time of two hours. You'll be required to turn in your exam after two hours.
- As a reference in the exam, we'll provide the operator precedence table and descriptions of select methods for these Java classes: `Math`, `Scanner`, `String` and `Random` classes.
- You may not use textbooks, notes, neighbors, calculators, or other electronic devices on the exam.
- **Your UW ID is required to take the exam.** Before the exam begins, drop off your UW ID at the front of the room and pick up a scantron form. On that form, fill in your name, UW ID number and other information as described on the exam. Sit in columns with an empty seat or an aisle to your right and left. When you're done, turn in your exam and scantron form, and pick up your UW ID.

A good exam taking strategy is to first focus on those questions that you feel most confident answering and skip over those that you're unsure about. Then, go back and answer those questions that you feel some confidence about, and repeat this process leaving till the end those questions that are most challenging for you. Try not to be stubborn or get stuck on a question. If you're having difficulty, go on to the next question. As you do questions make sure to write your work in the exam booklet so that you can verify it later when checking your answers.

The primary instruction for the exam is:

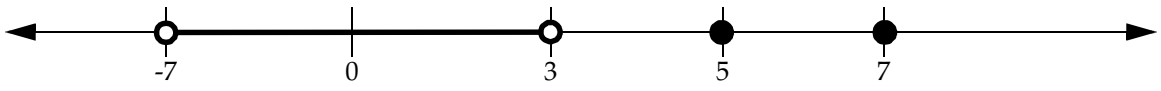
Choose the one best answer after reading all of the choices.

Questions will have only one best answer but will have other choices that, at first glance, might seem correct. Carefully consider all answers before making your choice. Then, mark your choice for the corresponding question number on your scantron form using a #2 pencil. Bring several #2 pencils to the exam.

Exam questions will test for your knowledge of concepts and terms (see #2, 4, 9), code reading/tracing (see #3, 5, 6, 7, 8, 10, 11) and also code writing (see #1, 5, 6, 7, 12).

Sample Exam Questions:

1.) Consider the following number line showing a specific range where the line is bold. Filled circles represent points that are included in the range and hollow circles represent points that are not included.



Which one of the following conditions results in `true` if `n` is in the range shown, and `false` otherwise?

A. `n > -7 || n < 3 && n == 5 && n == 7`
B. `n > -7 || n < 3 || n == 5 || n == 7`
C. `n > -7 && n < 3 && n == 5 || n == 7`
D. `n > -7 && n < 3 && n == 5 && n == 7`
E. `n > -7 && n < 3 || n == 5 || n == 7`

2.) Which one of the following statements about variables in Java is *false*?

A. A variable's memory location stores a single value.
B. A variable's name is associated with a memory location.
C. Variables can be declared in a method's body and in a method's parameter list.
D. It is a good programming practice to name variables based on their intended use.
E. It is a good programming practice to use an uppercase for the first letter in a variable's name.

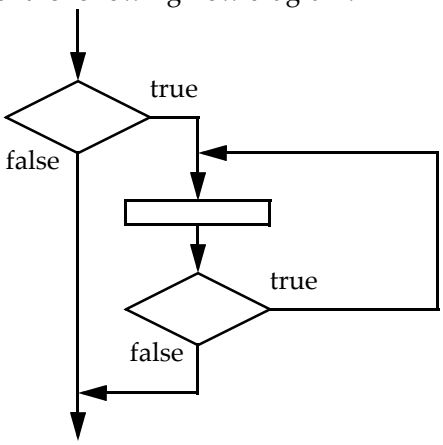
3.) Consider the following code fragment:

```
for (int j = 1; j <= 3; j++) {
   for (int k = j; k > 0; k--) {
      System.out.print(k);
   }
   System.out.println();
}
```

Which one of the following shows what is displayed by this code fragment?

A.
```
1
21
321
```
B.
```
1
21
321
4321
```
C.
```
121321
```
D.
```
1
12
123
```
E.
```
121
321
```

4.) Consider the following flow diagram:



This flow diagram represents which one of the following?

A. a while loop with an if statement nested inside
B. a while loop with an do-while loop nested inside
C. an if statement with an if statement nested inside
D. an if statement with a do-while loop nested inside
E. an if statement with a while loop nested inside

5.) Consider the following expression:

```
!(n > 11 && n <= 22)
```

The expression above is equivalent to which one of the following?

A. `n < 11 || n >= 22`
B. `n <= 11 || n > 22`
C. `n > 11 || n <= 22`
D. `n < 11 && n >= 22`
E. `n <= 11 && n > 22`

6.) Consider the following code fragment:

```
if (i == 11 || i == 22) { d = 5.5; }
if (i == 33)            { d = 7.7; }
d = 1.1;
```

For all values of `i`, this fragment has a result that is the same as which one of the following?

A.
```
switch (i) {
   case 11: d = 5.5; break;
   case 22: d = 5.5; break;
   case 33: d = 7.7; break;
   default: d = 1.1;
}
```

B.
```
switch (i) {
   case 11:
   case 22: d = 5.5; break;
   case 33: d = 7.7; break;
   default: d = 1.1;
}
```

C.
```
if (i == 11 || i == 22) { d = 5.5; }
if (i == 33)            { d = 7.7; }
else                    { d = 1.1; }
```

D.
```
if (i == 11 || i == 22) { d = 5.5; }
else if (i == 33)       { d = 7.7; }
else                    { d = 1.1; }
```

E.
```
d = 1.1;
```

7.) Consider the following incomplete method:

```
public static int factorial (int n) {
   int factorial = VALUE;
   for (int i = 1; CONDITION; i++) {
      STATEMENT
   }
   return factorial;
}
```

Which one of the following replacements for VALUE, CONDITION and STATEMENT could be used to complete the method so that it computes the factorial of n? The factorial of n is computed by multiplying all of the numbers between 1 and n (inclusive).

A. `VALUE: 1  CONDITION: i <= n  STATEMENT: factorial *= n;`
B. `VALUE: 0  CONDITION: i <= n  STATEMENT: factorial *= i;`
C. `VALUE: 1  CONDITION: i <= n  STATEMENT: factorial = factorial * i;`
D. `VALUE: 0  CONDITION: i < n  STATEMENT: factorial = factorial * n;`
E. `VALUE: 1  CONDITION: i < n  STATEMENT: factorial = factorial * i;`

8.) Consider the following method where `s1` and `s2` are properly initialized `String` objects:

```
public static boolean doIt(String s1, String s2) {
   int i1 = 0, i2 = 0;
   while (i1 < s1.length()) {
      if (s1.charAt(i1) == s2.charAt(i2)) {
         i1++;
         i2 = 0;
      }
      else if (i2 < s2.length() - 1){
         i2++;
      }
      else {
         return false;
      }
   }
   return true;
}
```

Which one of the following best describes what method `doIt` does?

A. It always returns true.
B. It returns true if and only if `s1` and `s2` begin with the same character.
C. It returns true if and only if all of the characters in `s1` are found somewhere in `s2`.
D. It returns true if and only if all of the characters in `s1` are found in `s2` in the same position.
E. It returns true if and only if `s1` and `s2` are the same strings (i.e., have the same characters in the same positions).

9.) Which one of the following is *not* an example of a good use of comments in a Java program?

A. Comments at the beginning of a source file stating the author's name, the code version and/or creation date, and known bugs in the code for that file.
B. Comments after primitive variable declarations specifying the intended range of values when it is a subset of the data type's full range of values.
C. Comments after each line of code in a method explaining what that line of code does.
D. Comments before a method definition describing any parameters and the return value.
E. Comments before a loop stating what the loop is intended to do.

10.) Consider the following two code fragments (assume `stdIn` is a properly initialized `Scanner` object):

fragment 1

```
i = stdIn.nextInt();
while (i >= 1) {
   System.out.println(i);
   i = i - 1;
}
```

fragment 2

```
i = stdIn.nextInt();
do {
   System.out.println(i);
   i = i - 1;
} while (i >= 1);
```

Under which of the following circumstances will the two code fragments produce the same output?

i. An integer less than 1 is entered.
ii. The integer 1 is entered.
iii. An integer greater than 1 is entered.

A. *i* only
B. *ii* only
C. *iii* only
D. *ii* and *iii* only
E. *i*, *ii*, and *iii*

11.) Consider the following program:

```
public class Parameters {

   public static void main(String[] args) {
      int a = 1, b = 2, c = 3;
      a = messUp(b, a);
      System.out.println(a + b + c);
   }

   public static int messUp(int a, int b) {
      int c = 4;
      a++;
      b--;
      return a + c;
   }
}
```

Which one of the following shows what the `Parameters` program displays when executed?

A. `11`
B. `12`
C. `13`
D. `123`
E. `723`

12.) Assume you are given three integer variables: `i`, `j` and `k`. Which one of the following Java conditions ***best*** implements the idea of "in increasing order"?

A. `i < j && i < k && j < k`

B. `i+1 == j || j+1 == k`

C. `i < j < k`

D. `i+1 == j && i+2 == k`

E. `i < j && j < k`