



University of Maryland College Park

Department of Computer Science

CMSC131 Fall 2019

Exam #1 Key

FIRSTNAME, LASTNAME (PRINT IN UPPERCASE):

STUDENT ID (e.g. 123456789):

Instructions

- Please print your answers and use a pencil.
- This exam is a closed-book, closed-notes exam with a duration of 50 minutes and 200 total points.
- **Do not remove the exam's staple.** Removing it will interfere with the scanning process (even if you staple the exam again).
- Write your directory id (e.g., terps1, not UID) at the bottom of pages with **DirectoryId**.
- Provide answers in the rectangular areas.
- Do not remove any exam pages. Even if you don't use the extra pages for scratch work, return them with the rest of the exam.
- Your code must be efficient and as short as possible.
- If you continue a problem on the extra page(s) provided, make a note on the particular problem.
- For multiple choice questions you can assume only one answer is expected, unless stated otherwise.
- You don't need to use meaningful variable names; however, we expect good indentation.
- **You must write your name and id at this point (we will not wait for you after time is up).**
- You must stop writing once time is up.

Grader Use Only

#1	Problem #1 (Miscellaneous)	62	
#2	Problem #2 (Diagram)	80	
#3	Problem #3 (Guess)	58	
Total	Total	200	

Problem #1 (Miscellaneous)

1. (3 pts) Java compilers produce:
 - a. Machine code (can run on the computer CPU)
 - b. Bytecode
 - c. None of the above.

Answer: b.

2. (3 pts) How many different combinations of 0's and 1's can be represented with 4 bits?
 - a. 4
 - b. 8
 - c. 16
 - d. None of the above.

Answer: c

3. (3 pts) What is the output of the following code fragment?

```
if ("laura".compareTo("tom") <= 0) {  
    System.out.println("laura");  
} else {  
    System.out.println("tom");  
}
```

Answer: **laura**

4. (3 pts) When is the body of a **do while** never executed?
 - a. When the appropriate condition is provided.
 - b. Never

Answer: Never

5. (3 pts) Provide a statement equivalent to `x *= 100;`

Answer: `x = x * 100;`

6. (4 pts) Which of the following are variable names that will compile? Circle all that will compile.

Globe7 telephone# car\$cos auto&boat

Answer/Grading:

- (1 pt) for indicating Globe7 will compile
- (1 pt) for indicating telephone# will NOT compile
- (1 pt) for indicating car\$cos will compile
- (1 pt) for indicating auto&boat will NOT compile

7. (4 pts) Which of the following are **reserved** names in Java? Circle all that apply.

else while place return

Answer/Grading:

- (1 pt) for indicating else is reserved
- (1 pt) for indicating while is reserved
- (1 pt) for indicating place is NOT reserved
- (1 pt) for indicating return is reserved

8. (4 pts) Write the octal representation of the decimal number 10.

Answer: 12

9. (4 pts) What is the output of following code fragment?

```
int a = 20;
int b = a++;

System.out.println(a + "," + b);
```

Answer: 21,20

10. (4 pts) Complete the following assignment so we are able to print the message below. Notice that double quotes surround Bob.

```
String message =
System.out.println(message);
```

Robert"Bob"Smith

Answer: "Robert\"Bob\"Smith"

11. (4 pts) Define a String constant named BEST_SCHOOL that has as value "UMCP".

Answer: `final String BEST_SCHOOL = "UMCP"`

12. (6 pts) Assuming **a**, **b** and **c** are integer variables, complete the following assignment that will initialize variable **decreasing** with the value true if **a**, **b** and **c** are in **decreasing** order, and false otherwise. For example, 10, 7, 3 are in decreasing order; 3, 5, 21 are not.

```
boolean decreasing =
```

Answer: `boolean decreasing = a > b && b > c;`

13. (7 pts) What is the output of the following code fragment?

```
int a = 10, b = 3;
boolean answer = ((--a >= 10) && (--b == 20));
System.out.println(a + "," + b + "," + answer);
```

Answer: 9,3,false

14. (10 pts) Re-write the following code fragment using a *for-loop*. The body of the for loop can only have the System.out.println statement (no other statement).

```
int a = 1, b = 4, limit = 10;

while (a <= limit) {
    System.out.println(a + "," + b++);

    a += 3;
}
```

Answer:

```
for (a = 1; a <= limit; a += 3) {
    System.out.println(a + "," + b++);
}
```

Problem #2 (Diagram)

Complete the program below. The program reads a number representing the maximum number of rows associated with a triangle. In addition the program will read a symbol and how many instances of that symbol should be used while drawing the triangle. Once the symbol has been printed the specified number of instances, the program will use * (if necessary). Use the messages "Enter rows:", "Symbol:" and "Number:" to read the number of rows, the symbol, and instances, respectively. The following are examples of running the program you are expected to write (remember your program must work for other values). Underlined text represents input provided by the user.

Enter rows: <u>4</u> Symbol: <u>\$</u> Number: <u>5</u> \$\$\$\$ \$** ** *	Enter rows: <u>6</u> Symbol: <u>\$</u> Number: <u>9</u> \$\$\$\$\$\$ \$\$\$\$** **** *** ** *
--	---

One Possible Solution

```
public static void main(String[] args) {
    int row, col, number, maxRows, maxCols;
    String symbol;

    Scanner scanner = new Scanner(System.in);

    System.out.print("Enter rows: ");
    maxRows = maxCols = scanner.nextInt();

    System.out.print("Symbol: ");
    symbol = scanner.next();

    System.out.print("Number: ");
    number = scanner.nextInt();

    for (row = 1; row <= maxRows; row++) {
        for (col = 1; col <= maxCols - row + 1; col++) {
            if (number == 0) {
                symbol = "*";
            } else {
                number--;
            }
            System.out.print(symbol);
        }
        System.out.println();
    }
    scanner.close();
}
```

Problem #3 (Guess)

Complete the program below. The program will generate a random value (let's called it toGuess) between 1 (inclusive) and 100 (inclusive). You can use 100 as the seed value needed by the Random class. The program will proceed to read an integer value from the user and verify whether that value is the one the program generated (toGuess). The program will keep asking the user for a value as long as the user have not guessed the expected value (toGuess). If the user enters the expected value, the program will print the message "You won" and end. If the value is larger than toGuess, the program will print "Too high" and "Too low" if smaller (and will continue asking for another value). To read a value use message "Enter guess:". You must use the Random object to generate random values. The following is an example of running the program you are expected to write (remember your program must work for other values). Underlined text represents input provided by the user.

```
Enter guess: 3
Too low
Enter guess: 8
Too high
Enter guess: 6
You won
```

One Possible Solution

```
public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);

    Random random = new Random(100);
    int expected = random.nextInt(100) + 1, guess;
    boolean done = false;

    do {
        String message;
        System.out.print("Enter guess: " );
        guess = scanner.nextInt();
        if (guess == expected) {
            message = "You won";
            done = true;
        } else if (guess > expected) {
            message = "Too high";
        } else {
            message = "Too low";
        }
        System.out.println(message);
    } while (!done);

    scanner.close();
}
```