

CMPU 102-52 - Spring 2017
Review Questions for Final Exam
Exam will be given at 1 to 3 pm on Tuesday, May 23rd, in RH 203

Solutions are provided for some examples. For many, just try typing them in DrJava Interactions pane.

This review sheet provides examples of the type of questions that may appear on the exam.

Object Oriented languages and Java

1. Describe the functions of a constructor. **To initialize the values of global variables, particularly instance variables.**
2. T or F: Java is both a compiled and interpreted language. **TRUE**
3. Explain what is meant by the statement "Java is a portable language". **Java source code will run on any machine that has a JVM installed. The compiler is a much larger piece of code than the JVM, which is an interpreter of "byte code".**
4. Name the eight Java primitive types. **Too easy.**
5. Give 2 examples of static methods in the Java standard library. **Math.random, Integer.parseInt**
6. Name a static constant in the Java standard library. **Math.PI**
7. Be able to show, in the diagram of a class, the scope of variable and constant names declared at various locations (e.g., inside the body of a class but outside any methods, inside a method body, inside a compound statement body).
8. What is garbage collection in Java? **Clearing of memory on the heap when there is no longer a reference to an object.**

Basic Java Syntax

1. Correct the following code segments if there are errors. Explain your changes. **Write an answer and then try typing the lines in in DrJava to test your answer.**

```
String s = "1234567890";  
int i = s.length;  
int j = s.indexOf('5');  
int k = s;
```

2. What is the output of the following code segment? Explain your answer. **Write an answer and then try typing the lines in in DrJava to test your answer.**

```
String s = "string1";  
System.out.println(s.length());
```

3. What is the output of the following code segment? Explain. **Write an answer and then try typing the lines in in DrJava to test your answer.**

```
String s = "perseverance";  
String t = s.substring(4,6);  
String u = s.substring(3,8);  
String v = s.substring(0,9);  
System.out.println(t + u + v);
```

4. What is the output of the following code segment? Explain. **Write an answer and then try typing the lines in in DrJava to test your answer.**

```
String s = "abcd ";
String t = "  abcd ";
String u = " ab cd ";
System.out.println(s.trim());
System.out.println(t.trim());
System.out.println(u.trim());
```

5. Write a program that prompts for and accepts a telephone number in the form ddd-ddd-dddd (where d is a digit in the range 0 through 9) and prints it out in the following format: (ddd)ddd-dddd

```
public class Phone {

    public static void formatPhone() {
        String ph =
            javax.swing.JOptionPane.showInputDialog
            ("Please enter a phone number in the form ddd-ddd-dddd: ");
        String output = "(";
        for (int i = 0; i < ph.length(); i++) {
            if (i == 3) { output += ")";
                } else output += ""+ph.charAt(i); }
        javax.swing.JOptionPane.showMessageDialog(null, output);
    }

    public static void main(String[] args) {
        Phone.formatPhone();
    }
}
```

6. What does the following program output? **Write an answer and then try typing the lines in in DrJava to test your answer.**

```
public class DisplayMotto {
    public static void main (String[] args) {
        System.out.println("We will not go it\alone.");
    }
}
```

7. What does the method return type void indicate?

8. Write a single println() statement that displays the following output: **Write an answer and then try typing the lines in in DrJava to test your answer.**

```
JJ
Jenna
Hannah
```

9. Write a Java expression for the following math expression: **Supposed to be $(1 / (1 + x)^2)$**

$$\frac{1}{(1+x)^2}$$

10. What does the modifier public indicate? **The code is visible to the world.**

11. Indicate whether the following names conform to the Java naming conventions. For those that follow a convention, indicate what a reader would expect them to be: method, class, final data field, or variable. Some may fit more than one convention, some may fit no naming conventions. Explain your answers.

- | | | |
|--------------------------|---|--------------------------|
| a) issquare no | e) mySquare variable or method | i) Drawing class |
| b) lsSquare class | f) getToken variable or method | j) COLOR constant |
| c) is_square none | g) _getToken none | k) gettoken none |
| d) Square class | h) fillSquare variable or method | l) MAX_var none |

The underlined parts signify that a variable is more likely to be a noun and a method is more likely to be a verb.

12. What is the output of the following code segment? **Write an answer and then try typing the lines in in DrJava to test your answer.**

```
int j = 4;
++j;
System.out.println("j is "+j);
```

13. What is the output of the following code segment? **Write an answer and then try typing the lines in in DrJava to test your answer.**

```
int j = 4;
int i = 4;
System.out.println("j is "+ ++j);
System.out.println("i is "+ i++);
System.out.println("j is "+ j);
System.out.println("i is "+ i);
```

14. Suppose a, b, and c are int variables initialized respectively to 1, 2, and 3. What values are assigned to e, f, and g? **Write an answer and then try typing the lines in in DrJava to test your answer.**

```
int e = ++a;
int f = b++;
int g = ++c + c++;
```

15. Consider the following code segment:
For each of the questions a-d, write an answer and then try typing the lines in in DrJava to test your answer.

```
if (i==j) {
    System.out.println("A");
}
else if ((i % j) < 3) {
    System.out.println("B");
}
else if (i > (j - 1)) {
    System.out.println("C");
}
else {
    System.out.println("D");
}
```

a) if i is 9 and j is 4, what is the output? **Try it**

b) if i is 4 and j is 9, what is the output? **Try it**

c) if i is 5 and j is 6, what is the output? **Try it**

d) if i is 5 and j is 9, what is the output? **Try it**

16. Write a code segment that sets integer k to 3 if the integer m is less than 5 and j is less than 0, otherwise, k should be changed to 1 if m is not less than k.

```
if ((m < 5) && (j < 0))
    k = 3
else if (!(m < k))
    k = 1
```

17. Write a method to print the following pattern:

```
*****
*****
*****
*****
****
***
**
*
```

Assume the number of rows and columns are equal and that this number is taken as an input parameter to the method. After coding this, try to code a lowerRight and upperRight triangle.

```
public class UpperLeft{

    public static void printTriangle(int n) {
        String output = "";

        for (int i = n; i > 0; i--) {
            for (int j = i; j > 0; j--) {
                output += "*";
            }
            output+="\n";
        }
        System.out.print(output);
    }

    public static void main(String[] args) {
        UpperLeft.printTriangle(8);
    }
}

public class LowerLeft{
    public static void printTriangle(int n) {
        String output = "";

        for (int i = 1; i <= n; i++) {
            for (int j = 0; j < i; j++) {
                output += "*";
            }
            output+="\n";
        }
        System.out.print(output);
    }

    public static void main(String[] args) {
        LowerLeft.printTriangle(8);
    }
}
}}
```

```

public class LowerRight{

    public static void printTriangle(int n) {
        String output = "";

        for (int i = 0; i < n; i++) {
            for (int j = 0; j < n-i; j++) {
                output += " ";
            }
            for (int j = 0; j < i+1; j++) {
                output += "*";
            }
            output+="\n";
        }
        System.out.print(output);
    }

    public static void main(String[] args) {
        LowerRight.printTriangle(8);
    }
}

public class UpperRight{

    public static void printTriangle(int n) {
        String output = "";

        for (int i = 0; i < n; i++) {
            for (int j = 0; j < i; j++) {
                output += " ";
            }
            for (int j = i; j < n; j++) {
                output += "*";
            }
            output+="\n";
        }
        System.out.print(output);
    }

    public static void main(String[] args) {
        UpperRight.printTriangle(8);
    }
}

```

18. Transform the following while loop into:

- * a for loop (possibly other statements are needed too)
- * a do while loop (possibly other statements are needed too)

```

int i = 1;
while (i < 20) {
    i++;
    System.out.println(i);
}

```

```

for (int i = 2; i <= 20; i++) {
    System.out.println(i);
}

```

```

int i = 2;

```

```

do {
    System.out.println(i);
    i++;
}while(i <= 20);

```

19. What, if anything, is wrong with the following code fragment?

It is an infinite loop because count will never be <= 0.

```

count = 50; // change to count = 0; AND
while (count >= 0) { // change to count <= 50
    System.out.println(count); // OR
    count = count + 1; // change to count = count - 1;
}

```

20. Trace the execution of the following loops by detailing the output of each: **Write your answer and then copy and paste each statement into DrJava**

- a)

```
for (int i = 1; i < 10; i++) {
    System.out.println(i);
}
```
- b)

```
for (int i = 1; i < 10; i = i + 2) {
    System.out.println(i);
}
```
- c)

```
for (int i = 11; i < 10; i++) {
    System.out.println(i);
}
```
- d)

```
int i = 1;
do {
    System.out.println(i);
    i++;
} while (i < 10);
```
- e)

```
int i = 10;
do {
    i--;
    System.out.println(i);
} while (i > 10);
```
- f)

```
int i = 2;
while (i < 10) {
    System.out.println("Happy, happy");
    i++;
}
```
- g)

```
int i = 1;
while (i > 10) {
    System.out.println("joy, joy");
}
```

21. Using your intuition, describe what is wrong with the following code fragment. Rewrite this code so it produces correct output. **The else is matched with the wrong if due to lack of {}s.**

```

if (total == MAX)
    if (total < sum)

```

```

        System.out.println("total equals MAX and is less than sum.");
    else
        System.out.println("total is not equal to Max");

    if (total == MAX) {
        if (total < sum) {
            System.out.println("total equals MAX and is less than sum.");
        }
    }
    else {
        System.out.println("total is not equal to Max"); }

```

22. How many times must the new operator be used to declare an array of n objects? **n + 1**

23. What is the lower and upper index of an array of n elements? **0 to n-1**

24. Identify the local variables, parameters, and global variables (instance variables) in the following class:

```

class Question1 {
    private int one; //data members or data fields (global variables)
    private int four; //ditto

    public double myMethod ( int one ) { //local variable
        double two = 2; //local variable
        int three = 3; //local variable
        return one + two * four; //local variable, local variable, global variable
    }
}

```

25. Rewrite the following class, using the optional this wherever it is allowed.

```

class One {

    private int var1;
    private int var2;

    public One () {
        var1 = 20;
        m2(var1);
    }

    public void m2 (int x) {
        var2 = x * 2;
    }
}

class One {

    private int var1;
    private int var2;

    public One () {
        this.var1 = 20;
        this.m2(this.var1);
    }

    public void m2 (int x) {
        this.var2 = x * 2;
    }
}

```

26. Which of the following are valid declarations? Which instantiate an array object? Explain.

- a) `int primes = {2, 3, 4, 5, 7, 11};` **invalid, should be `int[]`**
- b) `double[] elapsedTime = {11.47, 12.04, 11.72, 13.88};` **valid**
- c) `int[] scores = int[30];` **invalid, should be `"new int[30];"`**
- d) `int[] primes = new {2, 3, 5, 7};` **invalid, can't use {2,3,5,7} in this context**
- e) `int[] scores = new int[30];` **valid**
- f) `char grades[] = {'a', 'b', 'c', 'd', 'f'};` **valid**
- g) `char[] grades = new char[];` **invalid, no size given**
- h) `int[][] matrix = {{1,2,3,4},{2, 3}, {1,2,5,6,7}};` **valid**

27. Assume that `a` and `b` are one-dimensional `int` arrays that have been initialized and that `i`, `j`, and `k` are `int` variables that have been initialized.

- a) Write the statement that adds 6 to element `k` of `a`. **`a[k] = 6;`**
- b) Write the statement that copies element `i` of `b` to element `k+1` of `a`. **`a[k+1] = b[i];`**
- c) Write the statement that sums elements `i` and `k` of `a` and places the result in element `j+5` of `b`. **`b[j+5] = a[i] + a[k];`**

28. What is the output of the following code fragment? **Write your answer and then copy and paste statement into DrJava**

```
int[] a = new int[5];
a[0] = 0;
for (int i = 1; i <= 4; ++i) {
    a[i] = a[i-1] + i;
}
System.out.println(a[3]);
```

29. Given a two-dimensional array of `char`, defined and instantiated as an instance variable as follows,

```
char[][] map = { {'a','b','c','d'},
                 {'e','f','g','h'},
                 {'i','j','k','l'},
                 {'r','s','t'} };
```

Define a method to print the following:

Contents of map = a b c d e f g h i j k l r s t **use double nested for loops (Try it):**

```
System.out.println("Contents of map = ");
for (int i = 0; i < map.length; i++) {
    for (int j = 0; j < map[i].length; j++) {
        System.out.print(map[i][j]+" ");
    }
}
```

The exam will be cumulative but will stress material covered after exam 1.

- o Inheritance and polymorphism - know the concepts and how they are used in Java. Know how to draw and interpret inheritance diagrams.
- o Be able to define and demonstrate method overloading and overriding.
- o Know what exceptions are and how to throw and catch exceptions.
- o Know the difference between an interface and an abstract class and how these constructs facilitate inheritance and simulate multiple inheritance.
- o Know how casting works, both for primitive types and user-defined types in an inheritance hierarchy.
- * Stacks and Queues:
 - o Know the methods of the stack and queue ADTs as presented in lecture.
 - o Be able to trace the result of a series of stack or queue operations.
 - o Know how stacks are used in the Java Virtual Machine.
 - o Know how stacks and queues are implemented using ArrayLists and linked lists.
- * Inner classes:
 - o Know the difference between inner and anonymous inner classes. Be able to identify them.
- * Graphical user interfaces and event-driven programming:
 - o Be aware of the components we have used in GUIs and how to set up and add components to a window.
 - o Know the function of ActionListeners and how to make a JFrame a listener.
 - o Know how to define an ActionListener as an anonymous inner class.
 - o Be able to describe the difference between the flow of execution in a non event-driven and an event-driven program.

Review Questions:

1. What is the purpose of abstract classes, interfaces, and abstract methods?
2. What is the difference between the visibility modifiers public, protected, and private?
3. When is the use of the instanceof operator necessary?
4. When a method has more than one parameter, how does the Java compiler know which values go in which parameter?
5. Suppose we push n integers on a stack and then perform $n/2$ pop operations. What is the state of the stack after all these operations?
6. Suppose we enqueue n integers on a queue and then perform $n/2$ dequeue operations. What is the state of the queue after all these operations?
7. Suppose you wish to copy a queue into another queue, preserving the order of elements. Using only Queue operations, describe how this could be done.
8. Suppose you wish to reverse the order of elements on a pre-existing stack. Using only Stack operations, describe how this could be done. How many additional stacks are necessary?