1. (5 points) What is the output when the following Java codes are executed?

int x, y;

x = 3;

y = (++x) + (x++) \* (x++);

System.out.println("x = " + x);

System.out.println("y = " + y);

**Answer:**

x = 6, y = 24

1. (10 points) What is the output when the following Java codes are executed?

int[][] x = {{1, 2},{3, 4}};

int[][] y = {{5, 6},{7, 8}};

int[][] z = {{9, 10},{11, 12}};

z[0] = x[1];

z[1] = y[0];

x[1][1] = 100;

y[0][0] = 200;

System.out.println("z[0][0] = " + z[0][0]);

System.out.println("z[0][1] = " + z[0][1]);

System.out.println("z[1][0] = " + z[1][0]);

System.out.println("z[1][1] = " + z[1][1]);

**Answer:**

z[0][0] = 3

z[0][1] = 100

z[1][0] = 200

z[1][1] = 6

1. (10 points) Write a *static* void method called *doubleList* that takes an ArrayList of Strings as a formal parameter and replaces every String with two of that same String. For example, if the list stores the values ["how", "are", "you"] before the method is called, it should store the values ["how", "how", "are", "are", "you", "you"] after the method finishes executing.

**Hint**: the indices of the elements will be automatically updated if adding an element to the ArrayList. Check the lecture notes.

**Answer:**

public static void doubleList(ArrayList<String> array) {

for(int i=0; i<array.size(); i+=2) {

array.add(i, array.get(i));

}

}

1. [Section 6.1: 3] (5 points) Which of the following is the correct syntax to declare a Scanner to read the file example.txt in the current directory?
   1. Scanner input = new Scanner("C:\example.txt");
   2. Scanner input = new Scanner(new File("example.txt"));
   3. Scanner input = new File("\\example.txt");
   4. File input = new Scanner("/example.txt");
   5. Scanner input = new Scanner("C:/example.txt");

**Answer:**

b

1. [Section 6.1: 4] (5 points) Write code to construct a Scanner object to read the file input.txt, which exists in the same folder as your program.

**Answer:**

Scanner s = new Scanner(new File("input.txt"));

1. [Section 6.2: 8] (5 points) What is wrong with the following line of code?

Scanner input = new Scanner(new File(“C:\temp\new files\test.dat”));

**Answer:**

It has a .dat extension

1. [Section 6.3: 12] (10 points) For the next several questions, consider a file called readme.txt that has the following contents:

6.7 This file has

several input lines.

10 20 30 40

test

What would be the output from the following code when it is run on the readme.txt file?

Scanner input = new Scanner(new File("readme.txt"));

int count = 0;

while (input.hasNextLine()) {

System.out.println("input: " + input.nextLine());

count++;

}

System.out.println(count + " total");

input.close();

**Answer:**

6 total

1. [Section 6.3: 13] (10 points) What would be the output from the code in the previous question if the calls to hasNextLine and nextLine were replaced by calls to hasNext and next, respectively?

**Answer:**

input:6.7 input:This input:file input:has input:several input:input input:lines input:10 input:20 input:30

1. [Section 6.3: 14] (10 points) What would be the output from the code in the previous exercise if the calls to hasNextLine and nextLine were replaced by calls to hasNextInt and nextInt, respectively? How about hasNextDouble and nextDouble?

**Answer:**

0 total. Input:6.7 1 total

1. [Section 6.4: 20] (10 points) Write a *static* void method called *print* to print the following four lines of text into a file named message.txt, which exists in the same folder as your program:

Testing,

1, 2, 3.

This is my output file.

**Answer:**

public static void print() throws IOException{

PrintStream ps = new PrintStream(new File("message.txt"));

ps.println("Testing,");

ps.println("1,2,3.");

ps.println();

ps.println("This is my output file.");

ps.close();

}

1. [Chapter 6 Exercises: 6] (10 points) Write a *static* method called *readEntireFile* that accepts a scanner representing an input file as its parameter, then reads that file and returns its entire text contents as a String.

**Answer:**

public static void readEntireFile(String[] args) throws IOException{

Scanner s = new Scanner(new File("input.txt"));

while(s.hasNextLine()){

String x = s.nextLine();

System.out.println(x);

}

s.close();

}

1. [Chapter 6 Exercises: 8] (10 points) Write a *static* void method called *doubleSpace* that accepts a Scanner for an input file and a PrintStream for an output file as its parameters, writing into the output file a double-spaced version of the text in the input file. You can achieve this task by inserting a blank line between each line of output.

**Answer:**

public static void doubleSpace(Scanner Input, PrintStream out){

while(Input.hasNextLine()){

System.out.println(“Input.nextLine()”);

System.out.println();

}

}