CMPUT 114 - Midterm #2 Section A2 (A)

C. Jones - Fall Term 2000

Student Name:____ **Student ID #:** ______________________________

Instructions:

- You have **50 minutes** to complete this exam 🕓
- No books, notes, calculators or other aids are allowed in this exam 📖
- Please remove all hats, caps, and anything else that obscures your face ☺
- Please take appropriate measures to shield your answers from others around you 👁
- Once you leave the examination room, you will not be re-admitted ☹
- The exam has been printed single-sided so you can use the back of each page for your rough notes and calculations
- Note: Java is case sensitive, so your answers must be exact, unambiguous and precise. Please write clearly. Any illegible or ambiguous answers will be graded as zero. ☒
- Please write your answers in the spaces provided, and show any rough work that you feel is appropriate to help explain your answer ☑
- You must hand in **all** pages of this exam paper plus any other material distributed in the exam ✋
- Please write your name and student ID number in the box above ✍

Results:

Question #1 (tracing)	/12
Question #2 (Adventure Code)	/10
Question #3 (write the code)	/18
Question #4 (multiple choice)	/10
Total	**/50**

Please note: after the exams have been graded, 50% of the papers will be randomly selected, photocopied & retained

Question #1 - Tracing [12 marks]

Consider the following Java class, and then answer the question on the next page:

```
public class TextObject {

    public TextObject (String aString) {
        if ((aString.length() > 3) && (aString.substring(0,1).equals("1")))
             this.question = true;
        else
             this.question = false;
        this.label = aString;
    }

    public void second() {

       int loop;
       for (loop = 0; loop < this.label.length(); loop++)
             if (loop % 2 == 0)
                   System.out.print(this.label.substring(loop, loop + 1));
       System.out.println();
       System.out.println(this.question);
       System.out.println(TextObject.name);
       TextObject.name  =  TextObject.first(this.label);
    }

    private static String first(String aString) {

       String  anotherString  =  ""; // "" is an empty String
       String  theChar  =  "";

       while (aString.length() > 0) {
               theChar = aString.substring(0,1);
               anotherString  =  theChar.concat(anotherString);
               aString  =  aString.substring(1, aString.length());
        }
        return  anotherString;
    }
    private   boolean   question;
    private   String   label;
    private   static   String  name  =  "Fall";
}
```

Now consider the following application:

```
public class Fall00 {
      public static void main(String args[]) {
          TextObject text, word;
          text = new TextObject("abc");
          word = new TextObject("1234");
          text.second();
          word.second();
     }
}
```

The above program will produce 6 lines of output. Please write your answers in the table below, making sure that your answers are clear, legible and exact.

Output line 1:	
Output line 2:	
Output line 3:	
Output line 4:	
Output line 5:	
Output line 6:	

Question #2: Adventure Code [10 marks, 2 marks each]

Consider the code listing of the Room class, taken from the Adventure program Version 8. (This code is supplied separately from the exam paper. If you do not have a copy of the Room class, please ask a proctor to bring you one). When answering the following questions, make sure your answers exact and precise.

Questions a and b relate to the following line of code from the getRoomForAction method:

```
door = (Door) this.doors.elementAt(index);
```

a) In which class is the elementAt() method defined?

Answer

---

b) What is the declared type of the receiver object in this statement?

Answer

---

c) How many constructor invocations are there inside the Room class?

Answer

---

d) How many compound statements are there inside the buildMenu() method?

Answer

---

e) If we add the following invocation to the body of the enter() method, it would cause a syntax error. Why?

```
this.performAction("Quit", "adventurer");
```

Answer

Question #3 - Write the code [18 marks]

Consider a Java application that is intended to read an integer from the keyboard, and then display all of the even integers from the input integer down to zero. Here are some sample runs that illustrate how the program should work. The user's input has been shown in **boldface** in these example runs, so you can distinguish it from the program's output:

Sample run #1:

```
Enter an integer > 9
Evens: 86420
```

Sample run #2:

```
Enter an integer > 4
Evens: 420
```

Sample run #3:

```
Enter an integer > hello
abc
7.5
7
Evens: 6420
```

The following fragment of code invokes different methods in order to produce the behaviour shown above:

```
int input;
input = Fall.getInput();
System.out.print("Evens: ");
Fall.evens(input);
```

Your task is to write the methods that are be used by the above code to produce the behaviour shown in the sample runs. Note the following:

- The program should be able to handle invalid input (see sample run #3 above) by ignoring it and reading another piece of input, and continuing to do so until a valid integer is entered by the user
- Assume that the user's input will be positive (greater than or equal to zero) – do not worry about negative input.
- Assume that the Keyboard class has been added to your project folder.
- Your methods must be complete – i.e. you must write the method prototype as well as the method body
- Assume that the above code fragment and the methods are located in the same class (the Fall class).

You may find the following instance methods useful:

Class	Method name	Returns
Keyboard	readInteger( )	Returns a reference to an Integer that is represented by the String that contains all of the characters typed by the user until the ENTER key is pressed. If the text does not form a valid Integer, then returns null
Integer	intValue( )	Returns the value of this Integer as an int

Note: there is a public static final variable declared in the Keyboard class called in. The declared type of the in variable is Keyboard.

Write your code in the boxes on the next page:

Part a) Write the code for the getInput( ) method in this box (make sure that your method can handle invalid input): [10 marks]

Part b) Write the code for the evens( ) method in this box note: you must use a **while** loop: [8 marks]

Question #4 [10 marks] Please **circle** EXACTLY **ONE** choice as the best answer to each question [2 marks each]:

4.1) How many lines of output will be generated by the Java program segment (assume all declarations have been made correctly):

```
for (index = 0; index < 4; index++)
      for (count = 0; count < index; count++)
            System.out.println('*');
```

a. 6
b. 8
c. 4
d. 7
e. Infinite loop would occur
f. none – a syntax error would occur

4.2) Consider the following method prototype: `public static int aMethod(String aString, char aChar) {` Assume that `aMethod` is defined within a class called `Thing`. Assume that `var1` has been declared of type `int`, and `aThing` has been declared of type `Thing`. Which of the following is legal?

a. `var1 = Thing.aMethod("aChar", 'X');`
b. `aThing = aMethod("X", 'x');`
c. `var1 = aMethod("A", "A");`
d. `var1 = aThing.aMethod("X", 'X');`
e. `aThing = aMethod(" ", '2');`

4.3) A _________ statement can prevent "fall through" in a switch statement:

a. compound
b. iterative
c. break
d. functional
e. static

4.4) Which of the following describes white box testing?

a. The tester checks the outputs for each input against the expected outputs defined by the specification
b. The tester chooses inputs that exercise each statement or path in the code
c. Is used for functional testing to see if the software meets the specification
d. Is also known as "beta" testing and is conducted by a set of selected users after the black box testing has been completed
e. Focuses only upon iterative control structures to ensure that each loop terminates correctly

4.5) Black box test suites consist of:

a. Applets that act as a user interface to facilitate beta-testing
b. Compound statements which test all of the control structures in the code
c. Private static methods which exercise all of the instance methods in a class
d. Main programs that exercise all of the public methods
e. Public instance methods that exercise all of the constructors to ensure that object state is correctly initialized

This page has been left blank so you can use it for your rough notes and calculations