INSY 4305 ADVANCED APPLICATION DEVELOPMENT

ASSINGMENT 3

100 points

1. INSTRUCTIONS

- **Due date is March 15, 11:59 pm. Late submissions will get 0 points. NO EXCUSE!**
- **PLEASE, ONLY USE TECHNIQUES THAT WE LEARNED IN CHAPTER 5, CHAPTER 6, AND CHAPTER7.**
- In this assignment, you are expected to create **four** Java applications.
- Each question is **independent** of each other.
- You will **upload four Java files** on Blackboard. **If you do not upload .java files, your answers will not be evaluated.**
- **Do not forget to add comments to explain how your codes are working! Comments must be brief, clear, and understandable. Do not write long sentences!!!**
- **Write your codes individually! Do not copy of any of them from someone else!**

2. GRADING POLICY

- **Case 1:**
  - For each question:
  - I will compile your .java files. **If any compilation error occurs, 5 pts will be deducted.**
  - After that, I will check your algorithms whether they are correct or not. For example; if it says find odd and even numbers. I will check whether it really finds both even and odd numbers. **This part will be evaluated based on your work.**
  - Additionally, comments will be checked whether they clearly and briefly explain what you have done. **If comments are missing or not clear, enough, or brief 3 pts will be deducted.**

- **Case 2:**
  - For each question:
  - If there is not any compilation error:
    - I will try each case scenario stated in each question. For example; if it says find odd and even numbers. I will try both even and odd numbers. **This part will be evaluated based on your work.**
    - Additionally, comments will be checked whether they clearly and briefly explain what you have done. **If comments are missing or not clear, enough, or brief 3 pts will be deducted.**
- **Case 3:**
  - **If you do not upload a .java file, I will not evaluate your answer.**
- **Case 4:**
  - **If it is determined that you copy the codes from someone else, you will get 0 pt.**

QUESTIONS

Note: In each question, assume that the user enters correct inputs. You do not handle with exceptions.

1. **(35 pts)** Write an application that is including a three-question multiple choice quiz about Java programming language. Each question must have four possible answers (numbered 1 to 4). Also, ask user to type 0 to exit the test. [Assume that a user enters only integers 1,2,3,4, or 0].

Firstly, display a message that this quiz includes three questions about the Java programming language and display that each question has four possible answers.

If the answer is correct for the given question, display that the answer is correct. If the answer is not correct for the given question, display that the answer is wrong. If the user answers three questions correctly, display "Excellent", if two, display "very good", if one or fewer, display "It is time to start to learning Java". After that, ask user whether he or she wants to play again [you can use a boolean variable]. If user inputs true, start the game again. If user inputs false, then display a goodbye message and finish the game [Assume that user enters only true or false]. **UPLOAD MyTest.java**

Sample output:

```
Welcome to Java Programming Test
This test includes 3 questions
Each question includes 4 choices

Q1: Which of the following is not a Java primitive type?
1-Char 2-Byte 3-Real 4-Double (0 to exit the test)
Answer: 2
Answer is wrong

Q2: The format specifier ________ is a placeholder for an int value.
1-%d 2-%f 3-%s 4-%b (0 to exit the test)
Answer: 1
Answer is correct
```

```
Q3: To exit out of a loop completely and resume the flow of control at the nex
t statement after the loop, use a _______.
1-continue 2-break 3-return 4-any of the above (0 to exit the test)
Answer: 2
Answer is correct

Very good

Do you want to take the test again (True or False):false
Bye Bye
```

2. (**30 pts**). Use a one-dimensional array to solve the following program. Write an application that inputs six numbers [Keep the numbers in an array]. The numbers must be between 20 and 200, inclusive. If the number is not between 20 and 200, ask for another input. When a number is entered, display the number only if it is not a duplicate of a number already entered. If it is a duplicate of a number already entered, display a message that the number is a duplicate number and ask for another input. Provide for the worst case in which all six numbers are different. Display the complete set of unique values input after the user enters each new value. **UPLOAD Duplicate.java**

Sample output:

```
Enter number: 15
number must be between 20 and 200

Enter number: 21
21
Enter number: 22
21 22
Enter number: 23
21 22 23
Enter number: 22
22 has already been entered
21 22 23
Enter number: 24
21 22 23 24
Enter number: 24
24 has already been entered
21 22 23 24
```

```
Enter number: 25
21 22 23 24 25
Enter number: 26
21 22 23 24 25 26
```

3. **(35 pts).** Write a program that will help a student learn multiplication. Use SecureRandom object to produce two positive integers between 1 and 20. The program should then prompt the user with a question (use a sentinel-controlled loop), such as:

How much is 10 times 11?

The student then inputs the answer. If the answer is correct display the message “very good” and ask another question. If the answer is wrong to display the message “no, please try again” and let the student try the same question until the student finally gets it right.

You must create two java files: Multiplication.java and MultiplicationTest.java. In Multiplication add two methods: CreateQuestion and CheckResponse. In MultiplicationTest.java, test your methods. UPLOAD Multiplication.java and MultiplicationTest.java

Sample output:

```
How much is 15 times 13?
Enter your answer (-1 to exit): 425
No. Please try again.
Enter your answer (-1 to exit): 426
No. Please try again.
Enter your answer (-1 to exit): 195
Very Good!

How much is 7 times 5?
Enter your answer (-1 to exit): 35
Very Good!

How much is 6 times 17?
Enter your answer (-1 to exit): -1
```