# ITAD 242 Java Programming

# Final Exam

Final Exam is a “hands on”, “open book” test. You can use any resources you wish during the test.

**Restrictions**

* Use of e-mail is not allowed during the test.
* Only workstations in the classroom can be used to take the test. You cannot use your own laptops.
* No flash drives are allowed in the computer.

Please log in into Canvas.

Find the Final Exam under Assignments -> Tests -> Final Exam.

Find the corresponding files in the Files section of Canvas in the FinalExam folder.

**After you completed your work:**

* Submit your **Netbeans Projects** as zip file.

**Test Exercises**

1. (10 pts.) Use the predesigned “TestArrayListMethods” project that has been provided for this exercise.
2. Write a method   
   ***public static ArrayList< Integer> sortMerge( ArrayList< Integer> a, ArrayList< Integer> b)***   
   that sorts two array lists individually and then merges them producing a new sorted array list. Each time, append the smallest unprocessed element from either array list, then advance the index. For example, if a is 16 4 1 9 and b is 4 9 7 9 11 then sortMerge returns the array list 1 4 4 7 9 9 9 11 16
3. Complete the main method in “TestArrayListMethods” in a way that it asks the user to enter values for the first list and second list. Each input ends when the user simply hits <Enter> without a number. Error checking is not required. Capture a test run for the following user inputs:

list 1: 123, 45, 68, 96, 12, 13, 11, 68, 123  
list 2: 99, 88, 77, 66, 68, 13, 12 , 11

**You are required to:**

1. **Write/Complete the code for the class***TestArrayListMethods***. *// Make sure to document your code***
2. **Correct errors in the provided project** (for example import the needed packages)
3. **Submit your complete NetBeans project including the captured test run as text file.**
4. (10 pts.) Use the predesigned “ConsoleCalculator” project that has been provided for this exercise.

Implement the following functionality:  
The program asks the user to give number 1 and reads in number 1

then asks the user to give number 2 and reads in number 2

then asks the user to give the operation wanted: 'a','A', '+' for add, 's', 'S', '-' for subtract and reads this in

Implement the decision what to calculate with a switch statement

Make each input that is wrong throw an IOException with different message

Catch IOException and decide depending on the message what to tell the user

Loop up to 3 times until the user give the correct input, stop after the third try

Do the calculation and print the result

print finally "Goodbye" in any case.

**You are required to:**

1. **Write/Complete the code for the class**ConsoleCalculator**. *// Make sure to document your code***
2. **Correct errors in the provided project** (for example import the needed packages)
3. **Submit your complete NetBeans project including the captured test run as text file.**
4. (Total 40 pts.) Use the predesigned “Quiz” project that has been provided for this exercise.
5. (10 pts) Add a class NumericQuestion to the question hierarchy of Section 9.1. If the response and the expected answer differ by no more than 0.01, then accept the response as correct.
6. (10 pts) Add a class FillInQuestion to the question hierarchy of Section 9.1. Such a question is constructed with a string that contains the answer, surrounded by \_ \_, for exam­ple, " The inventor of Java was \_ James Gosling\_". The question should be displayed as   
   The inventor of Java was \_\_\_\_\_
7. (10 pts.) Add a class MultiChoiceQuestion to the question hierarchy of Section 9.1 that allows multiple correct choices. The respondent should provide all correct choices, sepa­rated by spaces. Provide instructions in the question text.
8. (5 pts.) Modify the checkAnswer method of the Question class so that it does not take into account different spaces or upper/ lowercase characters. For example, the response " JAMES gosling" should match an answer of " James Gosling".
9. (5 pts) Correct errors in the provided “Quiz” project and extend the main program to ask the user one additional question for each of the categories a)-c) above.

**You are required to:**

1. **Write/Complete/Correct the code for the classes**Quiz, Question, ChoiceQuestion, NumericQuestion, FillInQuestion, MultiChoiceQuestion**. *// Make sure to document your code***
2. **Correct errors in the provided project** (for example import the needed packages)
3. **Submit your complete NetBeans project as zip file.**
4. (Total 20 pts.) Use the predesigned “CarsMover” project that has been provided for this exercise. The project compiles and runs as is. It shows one unpainted car moving from left to right.
5. (5 pts) Add an additional constructor to the Car class that takes as input parameters: X and Y coordinates, and bodyColor.
6. (15 pts) Modify the main program to add a red and a blue car below the existing car. The blue car needs to move from right to left and the red car moves from left to right. Place the cars so that they don’t collide.

**You are required to:**

1. **Write/Complete/Correct the code for the classes**CarsMover, Car and CarComponent**. *// Make sure to document your code***
2. **Submit your complete NetBeans project as zip file.**
3. (Total 20 pts.) Implement a “Queue” class whose add and remove methods are synchronized. Write a main program “QueueDemo” that makes use of the Queue class in the following manner: Supply one thread, called the producer, which keeps inserting strings into the queue as long as there are fewer than 10 elements in it. When the queue gets full, the thread waits. As sample strings, simply use time stamps new Date(). toString(). Supply a second thread, called the consumer that keeps removing and printing strings from the queue as long as the queue is not empty. When the queue is empty, the thread waits. Both the consumer and producer threads should run for 100 iterations.

**You are required to:**

1. **Write the code for the classes**QueueDemo and Queue**. *// Make sure to document your code***
2. **Submit your complete NetBeans project as zip file.**