

INSY 4306 ADVANCED SYSTEMS DEVELOPMENT

ASSINGMENT -5

20 points

1. INSTRUCTIONS

- **For the late submissions, 5% of the total points will be deducted per hour automatically on Canvas.**
- You can only use the techniques that you have learned so far. Other techniques will not be accepted. **For other techniques, 10 pts will be deducted.**
- In this assignment, upload the following file:
 - **YourFirstNameYourLastName.zip /.rar [including AccountSavings.java, Person.java, PersonTest.java]**
- Each question is **independent** of each other.
- **Do not forget to add comments to explain how your codes are working!**
Short comments are acceptable.
- **Write your codes individually! Do not copy of any of them from someone else!**
- **NOTE: If you are using any IDE (Netbeans, Eclipse, etc.), please delete the statement **package xxxx;** (and save it again), from your application. Otherwise, I will get a compilation error, and you will lose 1 pt for each file giving a compilation error. It is your responsibility.**

2. GRADING POLICY

- **Case 1:**
 - For each question:
 - I will compile your .java files. **If any compilation error occurs, 1 pt will be deducted from each file including compilation errors**
 - 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.**
 - You can only use the techniques that you have learned so far. Other techniques will not be accepted. **For other techniques, 10 pts will be deducted.**
 - Do not drag and drop components. **It is not acceptable. You will get 0 point.**
 - 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 1 pt will be deducted.**
- **Case 2:**
 - For each question:
 - If there is not any compilation error:
 - I will try each case scenario as 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.**
 - You can only use the techniques that you have learned so far. Other techniques will not be accepted. **For other techniques, 4 pts will be deducted.**
 - 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 1 pt 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.

1. **(10 pts)** Create a class AccountSavings. The class has two instance variables: a double variable to keep annual interest rate and a double variable to keep savings balance. The annual interest rate is 5.3% and savings balance is \$10.

- Create a **void method** to calculate monthly compound interest.
- Create a method to run two threads. Use anonymous classes to create these threads. The first thread calls the monthly interest calculation method 4800 times. After that, the current thread sleeps 5 seconds. The second thread calls the monthly interest calculation method 1200 times. Before the main thread ends, these two threads must be completed. At the end, this method prints savings balance and annual interest rate.
- Add your main method into the same class and test your threads. Run your application 10 times. Each time the savings balance must remain the same.

2. **(10 pts)** Create a class Person. A person has a first name and a last name. Create a constructor for the person. Use the interface Runnable to create a thread from the class Person. This thread displays a message that the given person is started to work (display his/her first name and last name). Then, the thread sleeps 5 seconds. And then, the thread displays that the given person is completed its job (display his/her first name and last name).

Then, create a class PersonTest. Use executor services to create a thread pool with a size of 3. Then, submit 4 Person threads. In the main thread, display a message that the all tasks are submitted, and display that all tasks are done.