

# MPP Final Exam 4/21/16

---

Name: \_\_\_\_\_ StudentId: \_\_\_\_\_

There are a total of 45 points possible. Note there are questions on the back of this page. You have until 12:00 Noon to complete all the questions.

-----

- 1) We have a superclass Employee and a class Manager that extends Employee. The Manager class has a computeBonus method that is not in Employee.

We decide to add a new class that extends Employee called SeniorTeamLeader. The SeniorTeamLeader computeBonus is identical to Managers. Using new Java 8 Interface features create a class design so that the computeBonus() code is not duplicated in Manager and SeniorTeamLeader. Manager and SeniorTeamLeader still remain subclasses of Employee. Do not change the Employee class.

- A. (4pts) Show your UML class diagram for your solution.
- B. (6pts) Show code needed to implement the class diagram as follows –
- a. class declaration for Manager
  - b. class declaration for SeniorTeamLeader
  - c. all code for any new item you added in the class diagram.
  - d. describe any changes you made to the existing Manager class.
- 2) (4pts) If we attempt to run the following code what will be the result? What is an appropriate fix for this code, and what will be the result from the fixed code?

```
Stream<Integer> stream2 =  
    Stream.iterate(1, n -> n + 1);  
  
stream2.collect(Collectors.toList()).forEach(System.out::println);
```

- 3) (8pts) We have a list of undergraduate students. Students have the following attributes:

```
double gpa;  
String major;
```

Write a streams and lambda implementation that will generate a list of students called honorRoll that includes all the students that have a gpa greater than 3.0 and have a major of compSci.

- 4) (6pts) Suppose we want to sort the honoRoll list by gpa with highest gpa first. Show the changes you need to make to your lambda implementation and to the Student class for that case.
- 5) (4pts) We discussed two approaches for unit testing stream expressions in class. Describe those two approaches. Explain which approach would be best for testing your answer to problem 3

- 6) For the following interface:

```
Public interface Triplet<S, T, U> {  
    public S first getFirst();  
    public T second getSecond();  
    public U third getThird(); }  
}
```

- a. (5pts) Write a parameterized type implementation
- b. (5pts) Write a generic class implementation
- 7) (3pts) Employees is the superclass for the SeniorTeamLeader subclass. Will the following code to compile? If not, what change do we need to make to get it to compile?

```
List<SeniorTeamLeader> list1 = //populate with SeniorTeamLeader  
List<Employee> list2 = list1;
```