

## Problem Set 3 Grading Policy

- Graded problem sets are due Monday, October 3 at noon.
- Make sure to post all the grades on MIT server by noon. Otherwise, you won't be able to post grades online.
- Return graded hardcopies to the TA in charge.

### 1. Basic

- All errors should be marked and explained. Any resulting point deductions should be noted, with reason, as close as possible to the error.
- The final score for any problem in a problem set should not be less than zero.

### 2. Late or Incomplete Submission

- The problem set score is the score a student gets after her/his problem set has been graded normally.
- If the timestamp for a problem set is after Friday at 11:05AM, deduct 30 points from the **problem set score**. If a problem set is late, please do mark it as late.
- If the timestamp for a problem set is after the following Monday at 11:05, the final grade is automatically 0. However, you should still fully correct the problem set to provide feedback.

### 3. When Programs Don't Compile

- If the compilation error seems like it was unintended, take off 10 points. In general, it should be easy to decide whether a compilation error is unintended or not. For instance, sometimes students accidentally delete semicolons after they are done testing and running their programs, but before their final submission. These are the cases we are trying to capture. In these cases, it must be evident that the trivial addition of a semicolon or the insertion of a missing bracket should make the program work.
- Otherwise, take off 30 points.
- If you are wondering whether a particular compilation error is unintended or not, contact the TA.

### 4. When Programs Don't Execute Correctly

- When you execute a student's program, follow the control of the program. If you find a problem, fit it into one of the case descriptions below. Take off as many points as indicated, correct the mistake, and continue the execution of the program.

- Do not double penalize student for the same mistake. Once you have corrected their initial mistake, keep executing the program to find more mistakes.
- If you find additional mistakes that DO NOT fall into the same case, take off points for these.
- You should have a good and consistent justification for each error student makes.
- Be consistent throughout the entire problem set.

## 5. Formatting & Commenting

- In general, if a solution is unnecessarily difficult to understand due to poor formatting, bad variable names, or lack of comments, you should deduct 5 points at most from the problem set score.

## 6. Specific Case Description for PS 1

- Please give FULL credit to any part that runs and produces the correct results.
- Students can write the Java code the way they like, and it may be different from the official solution.
- `Movie.java`, `News.java`, `Series.java`, `Special.java` (9 Points per class; 36 points total)
  - 3 points: data member modifiers
  - 3 points: constructor
  - 3 points: getters and setters()
- `Duration.java` (15 Points)
  - 4 points: data members, constructor
  - 3 points: getters, setters
  - 8 points: (2 each) `getTotalInHours()`, `getTotalInMinutes()`, `getTotalInMinutes()`, `toString()`
- `DailySchedule.java` (45 Points)
  - 5 points: constructor
  - 4 points: getters, setters
  - 8 points: `dateToString()`. Do not penalize students if they have utilized the `toString()` method of `java.util.Calendar`, as long as the printed date is properly formatted, as a date and not a date and time. Method must be static.
  - 5 points: `channelNumberToName()`. Method must be static.
  - 8 points: `timeSlotToTime()`. Do not penalize students if they have chosen to implement only 12 hour or 24 hour notation. Be lenient with students who try to implement a strategy that may perform conversion to either format, if their code does not quite work.

- 15 points: `print()`. Actual output should be tabular and readable. Expect to see possibly only the title in the table. Also, titles, etc., within the table may be truncated. This is fine.
- `ProblemSet3.java` (4 Points)
  - 2 points: call `ProblemSet3.getDailySchedules()`
  - 2 points: call `DailySchedule print()` method on one of the `DailySchedule` objects in the array returned by `getDailySchedules()`