



香港浸會大學  
HONG KONG BAPTIST UNIVERSITY

## COMP2026/COMP2046 Practical Test 2

Quiz Date: 19/11/2022, 1:30pm-3:30pm

Duration: 120 minutes

### Instructions

- Submit a zip file of your source code to Moodle.
- You should ensure that all your answers are included in the zip file you have submitted. No submissions would be accepted after the end of the Practical Test.
- The duration for this Practical Test is 2 hours.
- This is an open books, open notes test. **You are not supposed to make any communication with other people or open any webpage during the test. Therefore, it is considered as violating exam regulation if you are using a phone or open Chrome/Edge/Firefox.**
- Total marks: 80. There are 20 marks here served as a bonus.
- All programs that you submit must contain comments with proper indentation.
- Follow the output of the program as given by the sample.
- You can use ArrayList when you feel that is appropriate.
- **Warning: cheating in practical test fail the course directly and your case will be recorded by the University.**

---

**Q1 Short question(20%)**

1. Write a class named `A1` that has a private integer array called `x`. Write other constructor for the class that takes an integer as a parameter. This parameter specifies the size of the array. In that constructor the values of all cells are initialized as 5;
2. Write a class named `A2` that has a final field of type `String` called `name`. The `String name` should be assigned by the parameter of its constructor. It also has a method called `getName` which returns the name of the field. All classes other than `A2` have no access to this field without using the method `getName`.
3. Write a class called `A3` that is a subclass of `A2`. Override the method `toString` so that it returns `"toString:"` + the name of the object `A3`. e.g., if the name of the object is Kevin, it returns `toString:Kevin`.
4. Write a class called `A4` that has public static void `main`. Inside `main`, create an array of ten `A2` objects. The first five objects are `A2` objects and the last five objects are `A3` objects. The name of the first five objects are `"A2"` + the index of the object (e.g. `A2-0`, `A2-1`,...`A2-5`). The name of the last five objects are `"A3"` + the index of the object (e.g. `A3-5`, `A3-6`, ...`A3-9`). Print out the name of each object in the array; one object per line.

We don't provide expected output nor starting code for this question. Make sure your program compiles. You will get at most half of the marks if your program does not compile.

A data file `data.txt` or `data2.txt` stores the coordinates of pixels of an image. The content of `data.txt` is as follow:

```

8 10
1 5 *
2 5 *
7 4 +
4 9 +
2 4 +
3 5 o
4 4 o

```

It represents the following image.

The diagram shows a 2D hexagonal lattice. A dashed rectangular boundary encloses a central region. Inside this region, the following symbols are placed in a grid-like fashion:

- Top row: \*
- Second row: +\*
- Third row: O
- Fourth row: O (left) and + (right)
- Fifth row: + (left)

The first line indicates the height (8) and the width (10) of an image. From second line onward, each line represent a coordinate (row, col) and the symbol that be printed at that coordinate. For example, the second line of the file is `1 5 *` which means row 1 (counted from top), column 5 (counted from left) is a `*` symbol.

**So basically same setting in Ptest1**, except you are working with the starting file [Canvas.java](#), [Point.java](#) this time.

You are also given the solution of Canvas of Ptest1. But believe me, it is less useful than you think.

---

### Q3 Happy Farm (25%)

There is a plant called RedBean. After a RedBean is planted, it is 0cm tall. It grows 1cm per day until it reaches 5cm. Then it will not grow anymore. Each RedBean has its own name, namely "RedBean1", "RedBean2", "RedBean3", and so on.

An allotment, or known as a farm, is a place where RedBeans are planted. It has an array of RedBeans. Because RedBean takes up only a very very tiny space, we can assume that an allotment can plant as many RedBeans as we want. The method `plantRedBean()` will plant a new RedBean to the allotment. The method `growRedBeans()` will make all RedBeans in the allotment grow 1cm. The method `checkRedBeans()` will print the number of RedBeans in the allotment and current status of the all RedBeans in the allotment. The method `harvest()` will harvest all RedBeans that is of full grown (5cm tall). The method will return the number of RedBeans that is harvested. After one RedBean is harvested, the allotment will have one less RedBean.

The details of the code can be referred to the program [Allotment.java](#). Your job is to complete the program and make the output of the program same as the following output.

```
Day 1
Red Beans in the allotment:3
RedBean1: 1
RedBean2: 1
RedBean3: 1
Day 2
Red Beans in the allotment:3
RedBean1: 2
RedBean2: 2
RedBean3: 2
Day 3
Red Beans in the allotment:3
RedBean1: 3
RedBean2: 3
RedBean3: 3
Day 4
Red Beans in the allotment:4
RedBean1: 4
RedBean2: 4
RedBean3: 4
RedBean4: 1
Day 5
Red Beans in the allotment:4
RedBean1: 5
RedBean2: 5
RedBean3: 5
RedBean4: 2
Day 6
Red Beans in the allotment:4
RedBean1: 5
RedBean2: 5
RedBean3: 5
RedBean4: 3
```

Day 7 - Harvest 3 beans

Red Beans in the allotment:1

RedBean4: 3

Plant two more beans

Red Beans in the allotment:3

RedBean4: 3

RedBean5: 0

RedBean6: 0

## Q4 Videos (25%)

This question is to model different types of videos. Study carefully of the driver program in [VideoTester.java](#) and also the result below.

The class Video is given to you as a sample. Your job is to create multiple classes and make the program compile. Write your answer in new Java files.

### Feature of Live Video

- Live Video cannot skip to the future. I can skip to the past (with negative value). After skipped to the past (rewind), I can skip to the future (with positive value).
- Live Video shows the relative time of the video played time (e.g. 1 second ago, 2 seconds ago, 3 seconds ago, etc.). When the video is played, the relative time **DOES NOT change**. It changes only when it is skipped.
- Live video does not have a fixed length, it ends only when it is stopped via the method `stop()`.
- It does not play if the live video stop.

### Feature of Ad Video

- Ad video cannot skip. It plays from the beginning to the end.
- It has a very similar toString method as Video class but added with the word (ad) at the end of the string.

```
OK
OK
OK
-----
How to code with Java in 1 hour (0/3600)
Skip -10 seconds
How to code with Java in 1 hour (0/3600)
Play one second, success? true
How to code with Java in 1 hour (1/3600)
Skip 5 seconds
How to code with Java in 1 hour (6/3600)
Play one second, success? true
How to code with Java in 1 hour (7/3600)
Skip -10 seconds
How to code with Java in 1 hour (0/3600)
Play one second, success? true
How to code with Java in 1 hour (1/3600)
Skip 20000 seconds
How to code with Java in 1 hour (3600/3600)
Video finished
Play one second, success? false
How to code with Java in 1 hour (3600/3600)
-----
Breaking News - COMP2045 exam paper leaked! ([live] 0 seconds ago)
Skip -10 seconds
Breaking News - COMP2045 exam paper leaked! ([live] 10 seconds ago)
Play one second, success? true
Breaking News - COMP2045 exam paper leaked! ([live] 10 seconds ago)
```

Skip 5 seconds  
Breaking News - COMP2045 exam paper leaked! ([live] 5 seconds ago)  
Play one second, success? true  
Breaking News - COMP2045 exam paper leaked! ([live] 5 seconds ago)  
Skip -10 seconds  
Breaking News - COMP2045 exam paper leaked! ([live] 15 seconds ago)  
Play one second, success? true  
Breaking News - COMP2045 exam paper leaked! ([live] 15 seconds ago)  
Skip 20000 seconds  
Cannot skip beyond the current time.  
Breaking News - COMP2045 exam paper leaked! ([live] 15 seconds ago)  
Play one second, success? true  
Breaking News - COMP2045 exam paper leaked! ([live] 15 seconds ago)  
Video finished  
Play one second, success? false  
Breaking News - COMP2045 exam paper leaked! (finished)  
-----  
Advertisement Clip (0/3) (ad)  
Ad Video can't be skipped  
Skip -10 seconds  
Advertisement Clip (0/3) (ad)  
Play one second, success? true  
Advertisement Clip (1/3) (ad)  
Skip 5 seconds  
Ad Video can't be skipped  
Advertisement Clip (1/3) (ad)  
Play one second, success? true  
Advertisement Clip (2/3) (ad)  
Skip -10 seconds  
Ad Video can't be skipped  
Advertisement Clip (2/3) (ad)  
Play one second, success? true  
Advertisement Clip (3/3) (ad)  
Skip 20000 seconds  
Ad Video can't be skipped  
Advertisement Clip (3/3) (ad)  
Video finished  
Play one second, success? false  
Advertisement Clip (3/3) (ad)