

CITY UNIVERSITY OF HONG KONG

Course code & title : CS2312 Problem Solving and Programming

Session : **Mock Paper for the Quiz**

Time allowed : 50 minutes

Contacts : Dr Helena WONG; Tel: (852) 3442-8042; Email: cshwong@cityu.edu.hk

Note:

1. There are **2** questions in this paper. Answer ALL questions.
2. Students may print the paper and write the answers in the blanks inside the paper; or students may view the paper and write the answers on blank white papers. Both pens and pencils are allowed.
3. For detail instructions please refer to the announcement on Canvas.

Remarks:

1. For all written code required by the questions, you should give precise JAVA code with proper programming styles. Marks may be deducted for redundant or unnecessary code.
 2. Apply best practices and skills of **object-oriented programming**.
 3. Pay attention to **code design and naming**. **Program execution speed is NOT** a major concern.
No need to handle input errors or exceptions unless explicitly mentioned in the question.
 4. If your answer is incomplete, you can still get partial marks based on the quality of the answer. Note: Do not remove or change any given code.
-

*This is a **closed-book** examination.*

*No materials or aids are allowed during the whole examination.
If any unauthorized materials or aids are found on a candidate during the examination, the candidate will be subject to disciplinary action.*

Question 1. (20 marks)

100 students are to take an online exam. To detect cheating, the teacher sets 100 versions for each question. **Canvas** then randomly assigns one different version for each student to answer. Without knowing this arrangement, some students may share their answers to others. This is considered as a serious act of **academic dishonesty**. Canvas can detect that easily, and reports to the Department.

Requirement

Now, you are given a program to complete (see next page). It is for a simplified scenario: only 3 versions of a question, and 3 students involved. Also, we assume that any sharer does not share a wrong answer (i.e. He only shares the correct answer for his question). The correct answer for each question is different from other questions.

You are given **main()** and the class **Question**, and the frameworks for classes **Student**, and **AnswerScript**. The program should give the outputs as shown in the comments within the square brackets [] inside **main()**. Your task: Add all required methods in **Student** and **AnswerScript** to achieve **Part A - C**.

Note: You should NOT add additional fields in the classes.

(a) [6 marks]

In Part A inside main(), objects for students and questions are created (each question object has the question text and the model answer); questions are received by students; then each student prints his question.

(b) [6 marks]

In Part B inside main(), students write their answers to produce the **AnswerScript** objects; then the answer scripts are dumped.

In the dumped output, we can see if any student has answered for a mismatch question. This evidence of cheating is checked in Part C.

(c) [8 marks]

In Part C inside main(), all answer scripts are checked for cheating cases.

For each cheating case detected, the program should print a line as shown in the comment in Part C inside main(). Otherwise, output **OK**.

Question 2. (15 marks)

Refer to last question. When the program runs, what will happen to the variables and what objects will be created? Please draw the memory picture to show all object variables in main() and all detail contents of the objects that they refer to. The detail contents should involve all *sub-objects* as well.

* You may ignore the parameter *args* of main()

* *sub-objects* means the objects that are included (referred via object fields) in the corresponding master objects. For example, a Student object has the name field for the name of the student, which is a string object that stores the spelling of the student name, therefore the name string object is a sub-object of the student object.

* For string contents, you may write "..." instead of the complete content. E.g., "Honest", "H...", "..."

```

public class Main {

    public static void main(String[] args) {

        // Part (A)
        // 3 questions are created: Each question has the question text and model answer
        Question q1v009 = new Question("Q1v009 question", "answer for Q1v009");
        Question q1v113 = new Question("Q1v113 question", "answer for Q1v113");
        Question q1v132 = new Question("Q1v132 question", "answer for Q1v132");

        // 3 student objects are created. Their names are passed to the constructor.
        Student student1 = new Student("Honest");
        Student student2 = new Student("Copier");
        Student student3 = new Student("Sharer");

        student1.receiveQuestion(q1v009);
        student2.receiveQuestion(q1v113);
        student3.receiveQuestion(q1v132);

        student1.printQuestion(); // Output [Honest should answer Q1v009 question]
        student2.printQuestion(); // Output [Copier should answer Q1v113 question]
        student3.printQuestion(); // Output [Sharer should answer Q1v132 question]

        // Part (B)
        AnswerScript as1, as2, as3;
        as1 = student1.write("wrong answer for Q1v009");
        as2 = student2.write("answer for Q1v132");
        as3 = student3.write("answer for Q1v132");

        as1.dump(); // Output [Honest answered "wrong answer for Q1v009" for "Q1v009 question"]
        as2.dump(); // Output [Copier answered "answer for Q1v132" for "Q1v113 question"]
        as3.dump(); // Output [Sharer answered "answer for Q1v132" for "Q1v132 question"]

        // Part (C)
        // For the following checkCheating method, the two parameters of checkCheating()
        // should mean the answer scripts to be checked in the relationship below:
        // first parameter: the sharer
        // second parameter: the copier

        AnswerScript.checkCheating(as3, as2); // Output [Suspected: Sharer => Copier. Report both to the Department.]
        AnswerScript.checkCheating(as1, as2); // Output [OK]
        AnswerScript.checkCheating(as1, as3); // Output [OK]
        AnswerScript.checkCheating(as2, as1); // Output [OK]
        AnswerScript.checkCheating(as2, as3); // Output [OK]
        AnswerScript.checkCheating(as3, as1); // Output [OK]
    }
}

public class Question {

    private String questionText;
    private String modelAnswer;

    public Question(String questionText, String modelAnswer) {
        this.questionText = questionText;
        this.modelAnswer = modelAnswer;
    }

    public String getModelAnswer() {
        return modelAnswer;
    }

    public String toString() {
        return questionText;
    }
}

public class Student {

    // the name of the student
    private String name;

    // the question that the student should answer
    private Question question;

    //Create and return the AnswerScript object
    public AnswerScript write(String ans) {
        return new AnswerScript(this, ans);
    }

    //Add required methods below
    //Note: You should not add additional fields in this class

}

public class AnswerScript {

    private Student writer;
    private String ans; //the answer

    public AnswerScript(Student s, String a) {
        writer = s;
        ans = a;
    }

    //Add required methods below
    //Note: You should not add additional fields in this class

}

```

Question 1

Write the Student class below:

Write the AnswerScript class below:

Question 2

Your answer: