Office Use Only

MONASH University

Sample
Examination Period

Faculty of Information Technology

EXAM CODES: **FIT2099**

TITLE OF PAPER: **OBJECT ORIENTED DESIGN AND IMPLEMENTATION – SAMPLE PAPER**

EXAM DURATION: 2 hours writing time

READING TIME: 10 minutes

THIS PAPER IS FOR STUDENTS STUDYING AT: (tick where applicable)

☐ Berwick	☑ Clayton	☑ Malaysia	☐ Off Campus Learning	☐ Open Learning
☐ Caulfield	☐ Gippsland	☐ Peninsula	☐ Monash Extension	☐ Sth Africa
☐ Parkville	☐ Other (specify)			

During an exam, you must not have in your possession any item/material that has not been authorised for your exam. This includes books, notes, paper, electronic device/s, mobile phone, smart watch/device, calculator, pencil case, or writing on any part of your body. Any authorised items are listed below. Items/materials on your desk, chair, in your clothing or otherwise on your person will be deemed to be in your possession.

No examination materials are to be removed from the room. This includes retaining, copying, memorising or noting down content of exam material for personal use or to share with any other person by any means following your exam. Failure to comply with the above instructions, or attempting to cheat or cheating in an exam is a discipline offence under Part 7 of the Monash University (Council) Regulations.

AUTHORISED MATERIALS

OPEN BOOK	**☑ YES**	**☐ NO**
CALCULATORS	**☐ YES**	**☑ NO**
SPECIFICALLY PERMITTED ITEMS **if yes, items permitted are:**	**☐ YES**	**☑ NO**

Sample Exam

Instructions to students

This is a SAMPLE EXAM intended as a study aid.

The questions are of a similar style to and cover some of the same topics as the real exam.

The exam is formatted similarly to the real exam.

There are six questions with a total of 79 marks available.

The number of marks available for a question should be a rough guide as to how long you spend on it.

If you need clarification on some part of a question, you can ask on the Moodle Discussion Forum or come to consultation.

Sample Exam

Question 1 – (6 + 6 = 12 marks)

You have joined a team that is working on a fully-featured version of the game similar to the one you worked on in your assignments, complete with a graphical user interface. It is to be released commercially. The plan is to re-use the engine code for similar games in the future.

The game runs reasonably well, but when you look at the engine code you discover that it contains a lot of "cut-and-paste reuse" – that is, repeated similar sequences of code, sometimes with a few very small changes.

a) Why do software engineers consider repeated code to be a bad thing? Write a paragraph or two about the risks of cut-and-paste reuse. (6 marks)
b) Describe how you would go about fixing the code to reduce these risks. (6 marks)

Sample Exam

Question 2 - (4 + 6 = 10 marks)

In mathematics, an integer is a number that is the member of the set {…,-2, -1, 0, 1, 2,…}

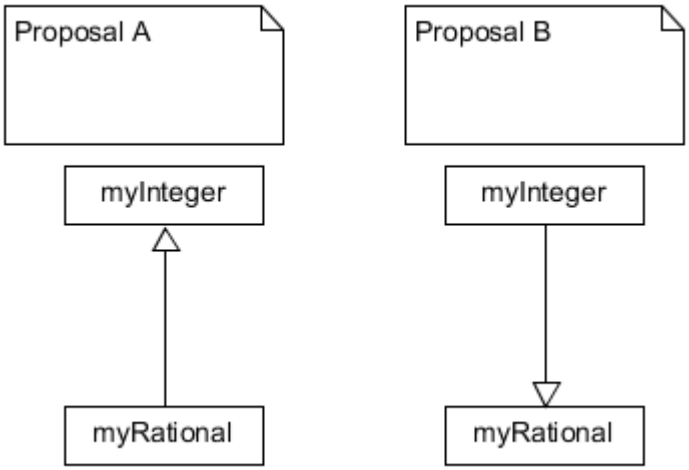
A rational number is any number that can be expressed as a fraction *p/q*, where *p* and *q* are integers and *q* is not 0.

Trivially, any integer *x* can be unambiguously expressed as a rational number x/1, so any integer is also a rational number.

Imagine you wished to define two classes, myInteger and myRational, that (initially) support the following operations:

- Constructing new objects with an initial value
- Incrementing the value by the value of another myInteger or myRational object.

Consider two class diagrams below that show proposed inheritance relationships between myInteger and myRational



a) In a couple of paragraphs, explain the advantages and disadvantages of both proposal A and proposal B, considering the design principles relating to inheritance discussed in FIT2099. (6 marks)
b) Propose an alternative design that avoids the disadvantages of both proposal A and proposal B. Draw a class diagram to illustrate it, and explain clearly how your proposal would implement the required operation. (6 marks)

Sample Exam

Question 3 - (4 + 5 + 6 = 15 marks)

Consider the following Java source code implementing a Unit class for a system similar to JavaUniversity from your labs:

```
/**
 * Represents a Unit at the university.
 *
 * @author Robert Merkel
 */
public class Unit {

	// the unit name
	private String name;

	// a list of assessments for this unit
	// invariant: the weight of all the assessments in this list must ALWAYS sum to
100.
	private ArrayList<Assessment> assessments;

	/**
	 *
	 * @param name the unit name
	 * @param code the unit code
	 * @param assessments the assessments for this unit. Their weights must sum to
exactly 100.
	 * @throws Exception if the assessment weights do not sum to 100.
	 */
	public Unit(String name, String code, Collection<Assessment> assessments) throws
Exception {
		this.name = name;
		int totalweight = 0;
		for(Assessment a: assessments) {
			totalweight += a.getWeight();
		}
		if (totalweight != 100) {
			throw(new Exception("Assessment weights do not sum to 100"));
		}
		this.assessments = new ArrayList<Assessment>(assessments);
	}

	/**
	 * Simple getter for unit name
	 * @return the unit name
	 */
	public String getName() {
		return name;
	}

	/**
	 * Simple getter for the list of assessments
	 * @return the assessment lists
	 */
	public ArrayList<Assessment>getAssessments() {
		return assessments;
	}
}
```

Sample Exam

a) The class and the class methods both have Javadoc comments. However, the attributes do not – despite the fact that you can annotate attributes with Javadoc. In your own words, briefly explain the reason for using Javadoc comments in your code. Does it matter much if the comments for the attributes *in this class* do not use Javadoc? Why or why not? (4 marks)

b) There is a potential design problem with getAssessments(). Explain what is wrong, and write an alternative version of getAssessments() that does not have this problem. (7 marks)

c) The Java language specification states that if a method throws a *checked exception* (that is, any exception that is not an instance of RunTimeException or a subclass), that method must either:
   - catch and handle them within its body; or
   - declare in its signature that it can throw them back to the caller (e.g. as the constructor for Unit does)

   Not all languages that support exception handling have this requirement. In Python, for instance, any method can raise any exception it chooses, and there is no requirement for the method signature to indicate this.

   In about a paragraph, explain *one advantage* and *one disadvantage* of each approach. (6 marks)

Sample Exam

Question 4: 2 + 2 + 4 + 4 = 12 marks

Design By Contract is an approach to software design that is supported directly in some languages, and which can be applied in others, such as Java, through the use of extensions or other language features. In Design By Contract, the designer of a class tells the clients of the class what the class does and what it needs via the *specification* of the class.

(a) Name two features of Java that allow a developer to write executable preconditions, and explain how these features can be used for this purpose. (2 marks)

(b) Why is it a good idea for a method not to try to cope with a violation of its preconditions, but simply to report the problem to the client that called it? (2 marks)

(c) Give an example of a method for which it would be much easier to write a postcondition than to write the body of the method. Explain when and how writing such a postcondition would aid in software development. (4 marks)

(d) Here is a simple Java method, in which the programmer would like to use a precondition. Choose a useful precondition for this method, explain what it is (in English), and write Java code to enforce the precondition (using standard Java, without using CoFoJa extensions). (4 marks)

```
/**
 * Calculate the arithmetic mean of a List
 * The arithmetic mean is defined as the sum of the values
 * divided by the number of values in the list.
 * @param nums - the list
 * @return the mean value
 */

public static double average(List<Double> nums) {
      // FIXME: precondition code should go here
      double sum=0.0;
      for(Double num: nums) {
            sum += num.doubleValue();
      }
      return sum / (double) (nums.size());
}
```

Sample Exam

Question 5: 10 marks

Below is some code from the act() method of a Droid in Star Wars game using the same rogue-like game engine as in your assignments.

Review this code for maintainability and extensibility.

For each design problem you identify:

a) Describe the problem clearly.
b) In a sentence or two, explain potential negative consequences of the problem.
c) Outline a potential fix for each design problem. You do not need to write actual code, but explain your fix in enough detail to make clear how your fix would address the problem. Write up to a few sentences.

Hint: consider "code smells"…

```
public void act() {

	// Code for other actions (omitted for brevity)

	// get a non-actor entity at this location, if any
	SWEntity potentiallyTakeable= getLocalEntity(this);
	if (potentiallyTakeable != null) {
	// if it's R2-D2
		if (this.model == 1) {
			SWAffordance takeaffordance =
			  getTakeAffordance(potentiallyTakeable);
			if(takeaffordance != null) {
				scheduler.schedule(takeable, this, 1);
				say("Bleepy bleep bloop");

			}
	 // if it's  C-3PO
		} else if (this.model == 2) {
			SWAffordance takeaffordance =
			 getTakeAffordance(potentiallyTakeable);
			if(takeaffordance != null) {
				scheduler.schedule(takeable, this, 1);
				say("I hope this doesn't get me into trouble.");

			}
		} else {
			say("Boss, there's something here!");
		}
	} else {
		// Do other things (omitted for brevity)
	}
}
	// Get the Take affordance for this entity, if any.
private SWAffordance getTakeAffordance(SWEntity entity) {
	for(SWAfforance aff : entity.getAffordances()) {
		if (aff instanceof Take) {
			return aff;
		}
	}
	return null;

}
```

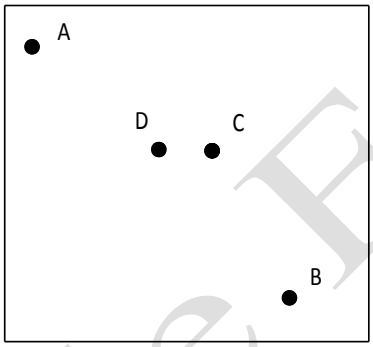
Sample Exam

Question 6: 5 + 3 + 5 + 7 = 20 marks

You're working on a system to monitor pairs of primary school students working on a geography assignment on the grounds of their school, which is too large for a teacher to keep all the students in sight at all times.

The students are supposed to work on the assignment in assigned pairs. Each student carries a smartphone with an app that periodically reports the position of that student to a monitoring system. The positions are reported as two numbers, x and y, which represent the offset in an easterly and northerly direction, respectively, in metres from an origin point in the center of the school. For instance, a student at the point (20.5, -5.3), would be located at a point 20.5 metres east and 5.3 metres south of the origin point.

To help teachers verify that they are working in their pairs, the system should regularly email the teacher a report listing each student and the student closest to them at that moment. For instance, if there are four students, A, B, C, and D, located as follows:

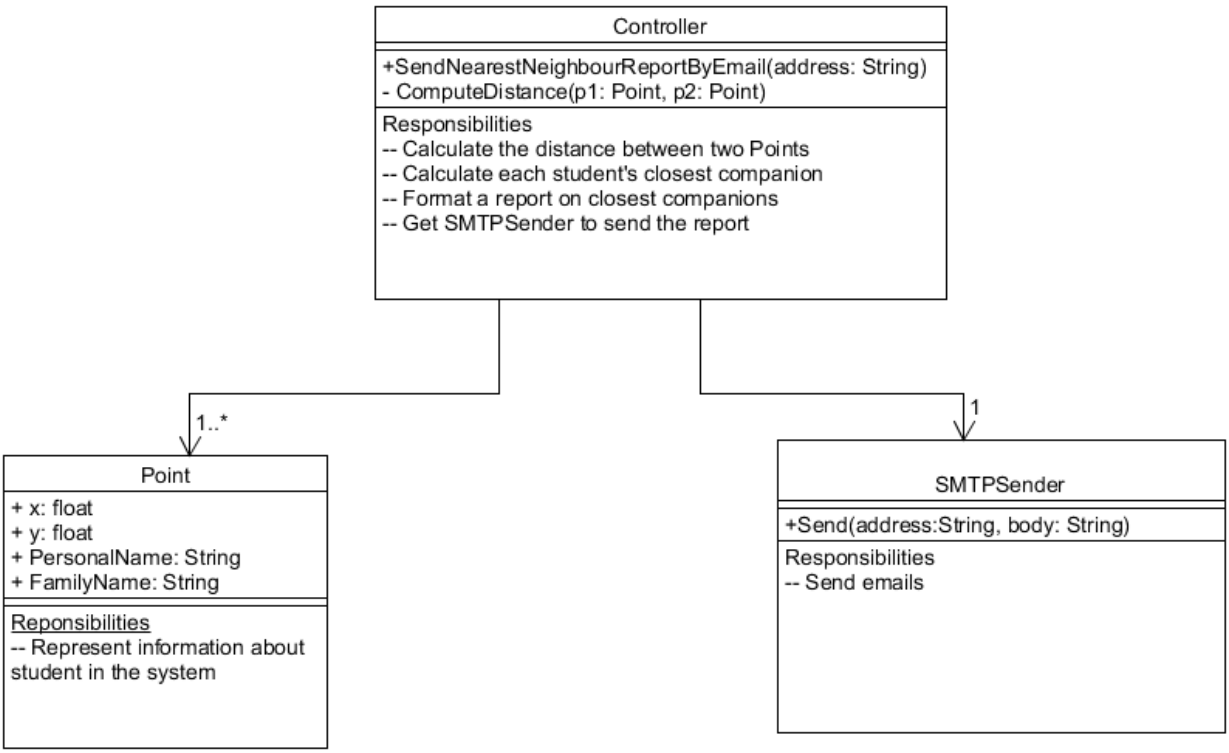


The emailed report should read:

A is closest to D
B is closest to C
C is closest to D
D is closest to C

You and a colleague are working on a design for the part of the monitoring system that emails the report. They send you the following class diagram, annotated with responsibilities for classes.

Sample Exam



a) Based on the description and the class diagram above, draw a sequence diagram for the "email nearest neighbour report". (5 marks)

b) How easy would it be to extend this system to support sending the report by other means? In a few sentences, explain the features of the design that make it easy or hard, and why they would do so. (3 marks)

c) Analyse the rest of the design in terms of the design principles we discussed in FIT2099. Write about half a page. (5 marks)

d) Suggest ways to improve the design, and explain why your changes are an improvement. Write a maximum of one page. Include a UML class diagram of your improved design. (7 marks)

Sample E

END OF SAMPLE EXAM