NATIONAL UNIVERSITY OF SINGAPORE

CS2103/T – Software Engineering

Practice exam- Part 2

Time Allowed: 20 minutes

Tip: Imagine you are in a project meeting with other team members, and you are drawing these diagrams on a whiteboard while explaining some aspect of the project to attendees. So, the speed of drawing is of essence; the neatness of the drawing doesn't matter. Do not worry about making the diagram neat, all lines straight, all shapes perfect etc.

These diagrams are mainly for you to help answer some questions in part 3. They will not be graded.

(a) Sketch a class diagram for the code given below. Follow the layout given. Also incorporate the following information into the diagram:

- An `Activity` object can be composed of other `Activity` objects i.e., sub-activities.
- A `Watcher` object may not be associated with more than 5 `Activity` objects.
- `UiWidget` class inherits the `ProgressWatcher`.

```
class Activity{
    private Watcher[] watchers = new Watcher[ProgressWatcher.MAX];
    //...
    void watch(Watcher w){
        //add w to watchers
    }

    Activity getInstance(){
        //...
    }
}
```

```
interface Watcher{
    void update(int value);
}
```

```
abstract class ProgressWatcher implements Watcher{
    static final int MAX = 10;
}
```

Activity

Watcher

ProgressWatcher

UiWidget

(b) Sketch an object diagram that has two `Activity` objects `a1` and `a2`, both being watched by a `UiWidget` object `u`. Furthermore, `a2` is a sub-activity of `a1`. Follow the given layout.

a1

u

a2

(c) Sketch a sequence diagram to illustrate the interactions resulting from the code below.

```
Client c = new Client(s, 100)
```

Code of the relevant classes are given below. Note that the two statements in the constructor execute in parallel to each other (the extra code required for parallelizing them is not shown for simplicity).

```
class Client {
    Client(Server s, int port) {
        init(s);      // parallel
        s.set(port);  // parallel
    }

    void init(Server s) {
        s.add(this);
    }
    //...
```

-- End of Part 2 –