The exam will be during your class session, on Tuesday, 1 October at 9:30am.
The exam will be open notes (hand written, typed, slides, programs), but closed Internet and readings. The exam will be submitted through your GitHub repository.

Content Covered
The exam will cover all material up to Wednesday, 25 September. The main topics covered are as follows.

• Broad understanding of an agent and a robot.
• See-think-act cycle.
• Sensors (classification, types, calibration).
• Localization and Navigation.
• Basics of leJOS and EV3 robots: movement, sensors.
• Lab 01 and 02 experiences.
• Readings.

Exam Format
The exam will contain 15 questions and will consist of a mixture of the following types of questions:

• Multiple choice with explanation.
• Short answer (provide 2-3 sentence explanation or provide an example).
• Code Analysis (given a line or a segment of code, describe what it does or provide missing functionalities).
• Code Writing (given a small task, write a few lines of code - nothing extensive - to implement the functionality).
Use the following to study for the exam:

- Slides.
- Class Exercises.
- Labs.
- Readings.

Sample Questions, NOT a Sample Exam:

1. Give an example of an intelligent agent and an example of a non-agent.

2. Give an example of an exteroceptive sensor. Explain why it is classified this way. What is its modality? Is it active or passive?

3. Calculate the accuracy of the ultrasonic sensor given its output is 2.3cm but the true value is 2cm.

4. Describe the steps of calibrating the temperature sensor.

5. Describe a scenario where odometry-based localization may not be appropriate.

6. What is the Markov property?

7. Given a code snippet below, explain what each of the five pose calls will (is supposed to) return.

```java
OdometryPoseProvider opp = new OdometryPoseProvider(pilot);
oppose(); // 1.
pilot.travel(60);
oppose(); // 2.
pilot.rotate(90);
oppose(); // 3.
pilot.travel(30);
oppose(); // 4.
pilot.rotate(-90);
oppose(); // 5.
```
8. Write Java statements to complete a navigation strategy of continuously moving in a square and a perception strategy of waiting for a touch sensor to be activated for the lego EV3 robot (programmed in leJOS) to stop and to play a sound. It should check for the touch sensor activation each time the robot completes its movement in a square. Assume an instance of the MovePilot class, called pilot has been declared and initialized.