LESSON PLAN	Grade: 12 G/A	Week: 2	Chapter 1: Programming
		Period: P1 – P2	Section 1: Arduino programming
		Time given: 2 x 45 min periods	

1) Learning outcomes

Essential LOs

- ☐ Employ control structures when writing codes.
- ☐ Differentiate between digital and analogue signals.
- ☐ Use Arduino to read and generate digital and analogue signals.

Strategies/Activities

Distance learning

- ★ LMS discussion forms
- ★ Microsoft teams group meetings, chats and assignments
- ★ E-Surveys
- ★ E-Polls

2) Keywords

- ▪ Digital signal
- ▪ Analogue signal
- ▪ PWM

3) Resources

- ▪ Computer
- ▪ Student Book
- ▪ Workbook
- ▪ LMS
- ▪ Microsoft teams
- ▪ Tinkercad
- ▪ Arduino IDE
- ▪ Video tutorials

4) Prior knowledge

- ▪ Electronics
- ▪ Introduction to Arduino

5) Assessment

Assessment for learning	Assessment as learning	Assessment of learning
★ Observations	★ Self-assessment	★ Workbook activities
★ Conversations	★ Peer-assessment	★ Test
★ Notes	★ Presentation	★ Quiz
★ Work sample	★ Graphic Organizer	
★ Checklist	★ Collaboration	
★ Diagnostics	★ Homework	

6) Starter

Distance learning

- ▪ Teacher to introduce students to the lesson aim at the beginning of the Microsoft teams lesson.
- ▪ Teacher to ensure all learning outcomes are outlined on LMS under the correct lesson.
- ▪ Use Microsoft teams screen sharing to show all learning outcomes and ensure understanding.
- ▪ Introduce the lesson's keywords through an LMS quiz or through an online game-based learning platform such as Kahoot or Quizlet.

7) Differentiation strategies

Differentiation to be met through designing various levels of activities. Some examples are outlined below:

Distance learning

- ★ LMS quizzes with visual aids MCQ, true or false and multi – response questions.
- ★ Differentiated LMS in class activities assigned to specific students
- ★ Reflection through LMS surveys or polls

Identify possible differentiation needs of the class based on prior classes and the starter activities.

8) Lesson activities

 Distance learning

Teacher led activities	Student led activities
☐ Teacher to start an online lesson on Microsoft teams.	☐ Students to log on to Microsoft teams lesson
☐ Teacher to upload reading materials to LMS prior to the lesson.	☐ Download any required material for the lesson from LMS.
☐ Students can view the books online via Al Diwan platform.	☐ Open the required book on Al Diwan.
☐ Teacher to recap on types of control structures.	☐ Activity: Students complete the revision activity.
☐ Teacher to create an LMS in-class activity or an online game such as Kahoot or Quizlet to refresh students' knowledge of types of control structures.	☐ Use Microsoft Teams to discuss the types control structures.
☐ Activity.	
☐ Teacher to explain the if-statement and for-statement.	☐ LMS activity: Students to complete in-class activities on LMS that cover activities 1.1.5 and 1.1.6 of the workbook.
☐ Teachers to create in-class activities on LMS of type fill in the blanks or multi-response to cover activities 1.1.5 and 1.1.6 of the workbook.	☐ LMS discussion form.
☐ LMS activities.	☐ Activity: Students complete the revision activity.
☐ Teacher to lead a class discussion on the difference between digital and analogue signals using LMS discussion forms.	☐ LMS activity: Students to complete in-class activity on LMS that cover activity 1.1.7 of the workbook.
☐ LMS discussion form.	
☐ Teacher to recap on Arduino's main functions to read and generate digital and analogue signals through an LMS quiz or through an online game-based learning platform such as Kahoot or Quizlet.	☐ Students should complete the activities as homework and submit them on LMS if not finished.
☐ Activity.	
☐ Teachers to create in-class activity on LMS of type fill in the blanks or multi-response to cover activity 1.1.7 of the workbook.	
☐ LMS activities.	
☐ Encourage questions using the chat box to ensure all students understand the lesson objectives.	

Workbook activities	**9) Cross-curricular links**
★ 1.1.5 – 1.1.7	 ▪ (Teacher to make cross-curricular links and activities where possible.)

10) Plenary

- ▪ Teacher to facilitate as students evaluate learning.
- ▪ Question pupils on what they have learned. Have learning outcomes been met? Has the lesson aim been achieved?
- ▪ Teacher to use LMS quiz or an online game-based learning platform such as Kahoot or Quizlet to check students understanding of the lesson objectives.
- ▪ Teacher to post homework activities through LMS and to assign a time frame for that for all unfinished activities.
- ▪ Teacher to create an LMS/Google/Microsoft surveys and share it with students where a checklist showing students' reflection will help self-reflect on the lesson for both teacher and students.
- ▪ Teacher to use LMS awards to give students recognition badges.

11) Reflection & Next steps

Activities that worked	Topics to be revisited

Activity 1.1.5

Edit the following code so that the LED turns on when the condition inside the if statement is not met.

```
int LED = 12;
int pushbutton = 8;
int buttonState;
void setup() {
pinMode(LED, OUTPUT);
pinMode(pushbutton, INPUT);
}
void loop() {
buttonState = digitalRead(pushbutton);
  if (buttonState == HIGH){
  digitalWrite(LED,HIGH);
  }
  else {
  digitalWrite(LED,LOW);
  }

}
```

Edit the following lines of code:

```
  if (buttonState == HIGH){
  digitalWrite(LED,LOW);
  }
  else {
  digitalWrite(LED,HIGH);
  }
```

Activity 1.1.6

Write an Arduino code that gradually increases the brightness of an LED from zero to its maximum brightness by an increment of 5. Then, gradually decrease it to zero again using 'for' loops.

```arduino
int brightness = 0;

void setup()
{
  pinMode(9, OUTPUT);
}

void loop()
{
  for (brightness = 0; brightness <= 255; brightness += 5) {
    analogWrite(9, brightness);
    delay(30); // Wait for 30 millisecond(s)
  }

  for (brightness = 255; brightness >= 0; brightness -= 5) {
    analogWrite(9, brightness);
    delay(30); // Wait for 30 millisecond(s)
  }

}
```

Activity 1.1.7

Write an Arduino code that turns an LED on when an analogue pin (A0) reads an analogue signal that is greater than 512.

```arduino
void setup() {
  Serial.begin(9600);
  pinMode(8, OUTPUT);
}

void loop() {
  // read the input on analog pin 0:
  int sensorValue = analogRead(A0);
  Serial.println(sensorValue);
  if (sensorValue > 512){
  digitalWrite(8, HIGH);
  }
  else{
  digitalWrite(8, LOW);
  }
}
```