



**CSARC**  
COMPUTER SCIENCE &  
ROBOTICS CERTIFICATION

**Certification in Arduino for Learners  
Examination Syllabus**

Updated: 22 Nov 2023

# Introduction

The **Certification in Arduino for Learners** aims to provide a standardised skill benchmark for Arduino-based microcontrollers. The focus of this certification is testing candidates on core skills in programming, electronics theory, breadboarding, and project making.

## Assessment Objectives Overview

### 1. Programming Theory

Candidates will be tested based on programming theory. They should understand the following concepts:

- 1.1 Programming Syntax
- 1.2 Variables and Mathematics
- 1.3 Data types (integer, long, float, unsigned)
- 1.4 Programming Logic (sequencing, selection, iteration)
- 1.5 Functional programming

### 2. Circuit Building

Candidates should be able to construct functioning circuits using the following:

- 2.1 Breadboard and Arduino connections
- 2.2 Digital and Analog Inputs/Outputs
- 2.3 I2C components

### 3. Practical Programming

Candidates should be able to demonstrate proficiency in:

- 3.1 Arduino-specific functions
- 3.2 Programming Functions
- 3.3 Utilising built-in and third-party libraries
- 3.4 User-defined functions



# Scheme of Assessment

## 1. Certificate Tiers

- 1.1 Each grade examination has a maximum possible score of 25.
- 1.2 Candidates will be issued a certificate if they meet the passing score requirement.
- 1.3 Certificates have different tier levels based on the score requirements as set out in **Table 1:**  
Certificate Tier Levels

**Table 1: Certificate Tier Levels**

Score Requirement	Certificate Tier
≥ 13	Pass
≥ 16	Merit
≥ 20	Distinction
≥ 23	High Distinction

## 2. Examination Format

- 2.1 The format of each grade examination will follow a score weightage as set out in **Table 2:**

**Table 2: Examination Format and Weightage**

Assessment Criteria	Score Weightage
Theory questions: <ul style="list-style-type: none"> <li>• Explaining programs</li> <li>• Error Spotting</li> <li>• Calculating value of variables</li> </ul>	5
Practical Circuit Building (3 marks each): <ul style="list-style-type: none"> <li>• Circuit for Practical Task 1</li> <li>• Circuit for Practical Task 2</li> </ul>	6
Practical programming (7 marks each): <ul style="list-style-type: none"> <li>• Program for Practical Task 1</li> <li>• Program for Practical Task 2</li> </ul>	14



# Certificate Examination Syllabus by Grade

## Grade 1

1.1	<b>C Prog Syntax</b> <ul style="list-style-type: none"> <li>• Semicolons</li> <li>• Open/close brackets or curly brackets</li> <li>• Capitalisation for functions</li> </ul>
1.2	<b>Variables and Mathematics</b> <ul style="list-style-type: none"> <li>• Correct declaration and assignment of variables</li> <li>• Logical use of Math functions to allow code to progress logically</li> </ul>
1.4	<b>Programming Logic</b> <ul style="list-style-type: none"> <li>• WHILE loops</li> <li>• If/Else</li> </ul>
2.1	<b>Breadboard and Arduino Connections</b> <ul style="list-style-type: none"> <li>• Power Rails properly utilised</li> <li>• Correct use of breadboard to connect circuit components together</li> <li>• Use of correctly-valued resistors</li> </ul>
2.2	<b>Digital or Analog Inputs/Outputs</b> <ul style="list-style-type: none"> <li>• LED</li> <li>• Buzzer (Active)</li> <li>• Push Button</li> <li>• Potentiometer</li> <li>• Light Dependent Resistor (10K)</li> </ul>
3.1	<b>Arduino Specific Functions</b> <ul style="list-style-type: none"> <li>• digitalWrite</li> <li>• digitalWrite</li> <li>• analogRead</li> <li>• analogWrite</li> </ul>
3.2	<b>Programming Functions</b> <ul style="list-style-type: none"> <li>• If/Else</li> <li>• While loops <ul style="list-style-type: none"> <li>○ Finite count while loops using counter variable</li> <li>○ Conditional while loops ending based on sensor state</li> </ul> </li> <li>• Variables</li> <li>• Operators(+, -, *, /)</li> </ul>



## Grade 2

1.2	<b>Data Types</b> <ul style="list-style-type: none"> <li>• Use of appropriate data types – Int or Long</li> </ul>
1.3	<b>Programming Logic</b> <ul style="list-style-type: none"> <li>• Repeat criteria</li> <li>• WHILE loops and FOR loops</li> <li>• Logical If-Else criteria</li> </ul>
2.2	<b>Digital or Analog Inputs/Outputs</b> <ul style="list-style-type: none"> <li>• LED (5mm)</li> <li>• Buzzer (Active)</li> <li>• Push Button (12mm)</li> <li>• Potentiometer</li> <li>• Light Dependent Resistor (10K)</li> <li>• 180-degree Servo Motor (MG90S/SG90S)</li> <li>• Ultrasonic Sensor (HC - SR04)</li> </ul>
2.3	<b>I2C Components</b> <ul style="list-style-type: none"> <li>• Liquid Crystal Display (1602 I2C LCD)</li> </ul>
3.1	<b>Arduino Specific Functions</b> <ul style="list-style-type: none"> <li>• digitalWrite</li> <li>• digitalRead</li> <li>• analogWrite</li> <li>• analogRead</li> </ul>
3.2	<b>Programming Functions</b> <ul style="list-style-type: none"> <li>• If/Else</li> <li>• FOR loops</li> <li>• Variables</li> <li>• Operators(+, -, *, /)</li> <li>• Millis() function</li> </ul>
3.3	<b>Libraries</b> <ul style="list-style-type: none"> <li>• Liquid Crystal Display (1602 I2C LCD)</li> <li>• 180-degree Servo Motor</li> <li>• Ultrasonic Sensor (Optional)</li> </ul>
3.4	<b>User Defined Functions</b>



## Grade 3

To be updated

