

code.sprint^{MT}

Preparatory Workbook 2026

Your starting point towards Codesprint Malta



Table of Contents

Welcome to CODESPRINTmt	2
Workbook Guidelines.....	3
Grading Distribution.....	4
Section A	5
Task 1: Hotel Safe Box.....	5
Task 2: Lotto Ticket	7
Task 3: Point-of-Sale System.....	8
Task 4: MasterMind	10
Assessment Criteria.....	12
Section B	14
Self-Reflection Exercise	14
Final Presentation Guidelines	15

Welcome to CODESPRINTmt

Join **Code.Sprint Malta**, a timed coding challenge that puts your problem-solving / coding skills to the test. Besides the coding fun, you will have the opportunity to meet coding enthusiasts and sharpen your coding competitive skills while having the chance to win amazing prizes.

Code.Sprint^{MT} is divided into two phases: 1) Qualifiers Round, and 2) Finals Round. The top 10 participants from the qualifiers round will pass on to the final round.

During both rounds:

- you will be given a task to complete within a fixed time.
- you are to carry out the tasks on an individual basis.
- a judging panel will give points to your work according to an established criteria rubric.
- a detailed assessment rubric will be given during the contest. Therefore, you will be aware of what the judges are expecting from your work.

Moreover, Code.Sprint^{MT} is **accredited at MQF Level 3**, thus offering you the opportunity to top up your list of academic achievements. You can be awarded either a **1 ECTS Non-Formal MQF3 Award** (not graded) or a **2 ECTS Applied MQF3 Award** which is graded as explained in section [Grading Distribution](#).

NON-FORMAL AWARD (1 ECTS)

1. Complete this preparatory workbook:
 - Develop solutions according to the tasks found in this workbook.
 - Do the Self-Reflective exercise once all the tasks have been completed.
 - Send your work (x4 source codes + self reflection exercise in PDF as ONE ZIP file).
2. Participate during the Qualifiers Round.
3. Submit the solution of all the tasks given during the qualifiers round to the judges.

APPLIED AWARD (2 ECTS)

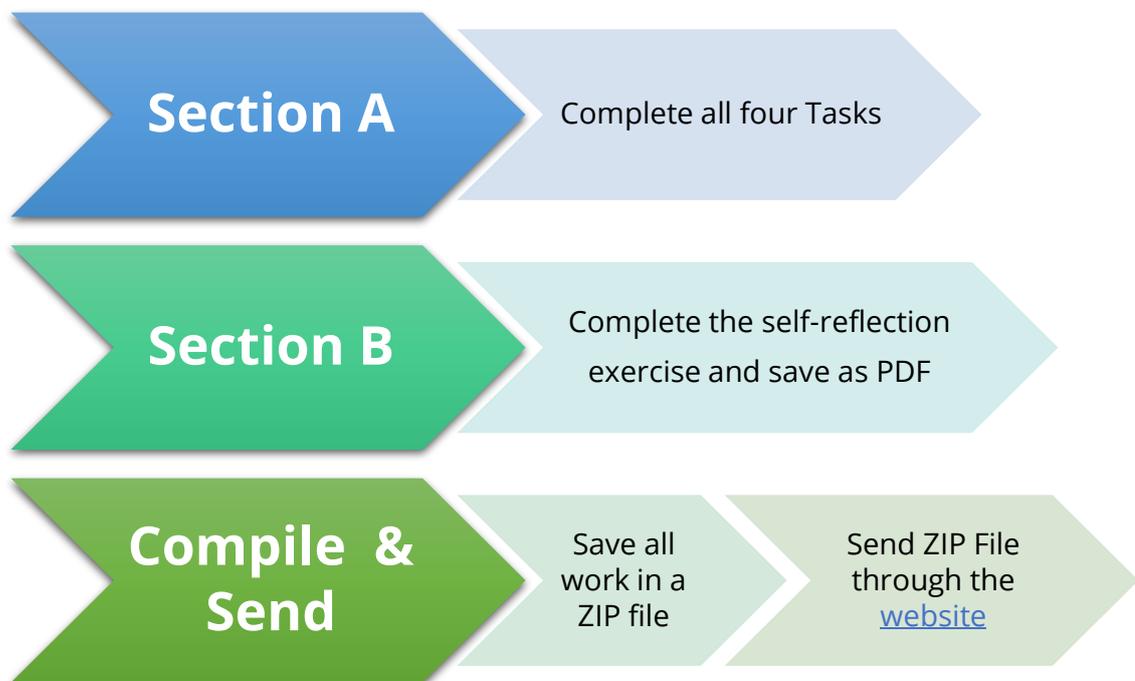
4. Qualify to the Final Round.
 5. Participate in the Bootcamp Session.
 6. Participate in the Final Round.
 7. Submit the solution of every task given during the final round.
 8. Create a presentation in any format you prefer of not more than 5 minutes.
 9. Demonstrate the presentation online a day or two prior to the Award Ceremony.
- (A graded award is given when all work is assessed by experts in the area)*

Workbook Guidelines.

This workbook includes:

- Four tasks in [Section A](#)
- [Assessment Criteria](#) for every task
- Self-Reflection Exercise in [Section B](#)
- Guidelines for the [Final Presentation](#) (*requirement for the 2 ECTS Applied MQF3 Award*)

Participants are expected to complete all the tasks presented in section A. Besides the development of the source code, participants are also expected to complete the self-reflection exercise in Section B.



ALL WORK MUST BE SAVED IN **ONE ZIP FILE** AND SENT BY THE CLOSING DATE AS INDICATED IN ACCREDITATION SECTION IN THE WEBSITE.

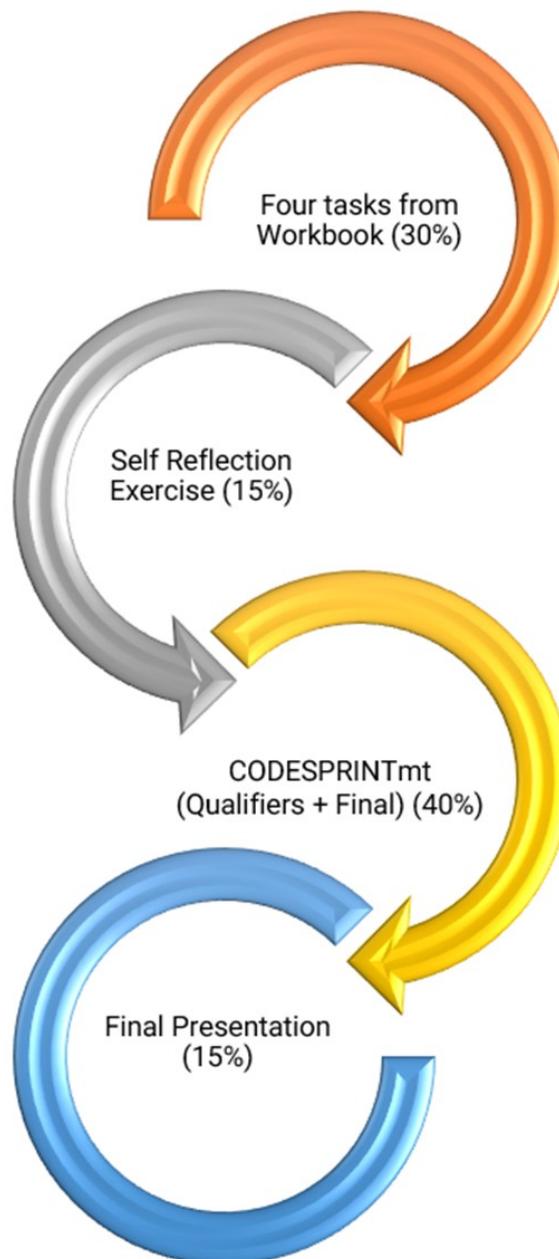
The Zip filename should be your name_surname_school.pdf
e.g. alexia_brincat_stignatiuscollege.zip

Grading Distribution

The 2 ECTS Applied MQF 3 Certification is graded as follows:

0 - 39	40 - 59	60 - 89	90 - 100
FAIL	PASS	MERIT	DISTINCTION

The marks from each phase of the event contribute to a global mark. The distribution of marks is as follows:



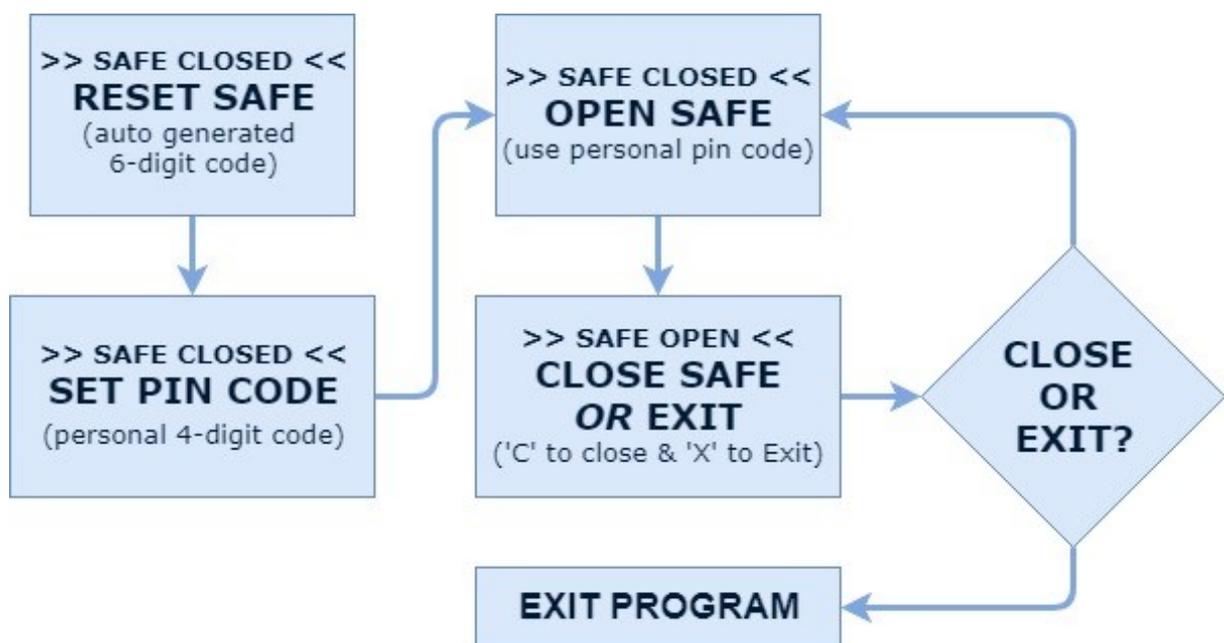
Section A

Task 1: Hotel Room Safe Box

Hotel guests should feel safe leaving their belongings in their room. Hotels from all around the world, nowadays offer a safe box in every room. Guests can therefore use the safe box to secure any valuable personal belongings.



The safety box has a keypad so that the user can use a personal pin number to open and close the safe. Write a program that simulates a safe box which works as follows:



Program Rules:

- The guest must reset the safe box by using a 6-digit random pin code which is auto generated and displayed upon initialization of the program. *A proper message is displayed if a wrong pin, or a non-6-digit pin is entered.*
- After resetting the safe box, the guest should enter a personal 4-digit pin. *A proper message is displayed if a non-4-digit pin is entered.*
- The user can open the safe box by entering the personalised pin code. *A proper message is displayed if a non-4-digit pin is entered.*
- The user can close the safe box by entering '**C**' or exit program by entering '**X**'. *A proper message is displayed if the character 'C' or 'X' is not used to close the safe box or exit program respectively.*
- Proper indications of the status of the safe box should be displayed with every step.

Name your program **task1_name.py**
(e.g. task1_john.py)

Task 2: Lotto Ticket

The lottery is a game in which players pay for a ticket, select a group of numbers and win prizes based on how they match the drawn results.



Write a program that simulates a lottery system according to rules below:

- The lottery system has a set of numbers available which is from 1 to 45.
- The lottery prize is that of €500,000.
- Five lottery numbers are randomly drawn and are not visible to the user.
Proper validation is required to avoid duplicate numbers.
- The user is asked to purchase a lottery ticket.
- A lottery ticket is made up of five non-duplicate numbers.
Proper validation is required to avoid non-valid and duplicate numbers.
- According to the numbers guessed, a prize is won:
 - With three numbers guessed, the user wins 10% of the lottery prize.
 - With four numbers guessed, the user wins 25% of the lottery prize.
 - With five numbers guessed, the user wins lottery prize in full.

The result screen should display the numbers drawn, the number of correct guesses, and the prize won (if applicable).

Name your program **task2_name.py**
(e.g. task2_john.py)

Task 3: Point-of-Sale System

POS systems have replaced most traditional cash registers due to their ability to connect to the retailer's main database system. Having all data stored and accessible within one system makes daily operations more efficient and more profitable.



Write a program that simulates a POS system. This system has a Log-In Screen and a Main Menu as shown below:

Log In Screen	
Option	Description
Cashier Log In	A cashier must be logged-in to proceed to the Main Menu. <i>(Details of the registered cashiers is shown in Table 2)</i>
Exit	Terminates the program.

Main Menu	
Option	Description
Enter new transaction	Allows the cashier to enter the items that the client wants to buy. <i>(A list of the stock items is shown in Table 1)</i>
Issue Receipt	This option displays the receipt of the last transaction carried out. <i>(A sample is shown in Figure 1)</i>
Display Stock List	This option displays the list of items that a client can buy. <i>(A list of the stock items is shown in Table 1)</i>
Cashier Sign Out	Allows a logged-in cashier to sign out and return to the Log-In Screen. <i>(Details of the current registered cashiers is shown in Table 2)</i>

Stock List	
Items	Price
Printer	€67.99
Monitor	€138.00
Keyboard	€12.50
Graphics Card	€114.99
Soundbar	€249.00
Hard Disk	€66.95
Headset	€17.55
Smartwatch	€135.00
Camcorder	€329.00
Drone	€449.99

Table 1

```

-----
<< EASY-SAVE SUPERMARKET >>
102, Flower Street, Filfla
*****
Sun Jan 20 11:48:45 CET 2019
Receipt: 66432020
Cashier: mallia.amy
*****
MONITOR                Eur 138.00
HEADSET                Eur 17.55
SOUNBAR               Eur 249.00

SUBTOTAL:              EUR 404.55
VAT:                   Eur 72.82
TOTAL:                 Eur 477.37

*****
                        FISCAL RECEIPT
                        THANK YOU
----- CUT-HERE -----

```

Figure 1: Sample Generated Receipt

Registered Cashiers	
Username	Password
borg.steve	Borg87max
zammit.rita	RitPopSing!
agius.john	ToyotaBeSt
vella.carlos.02	Rock!n!Roll
mallia.amy	\$Gaga\$Lady\$

Table 2

Name your program
task3_name.py
(e.g. task3_john.py)

Code Hint: The code below displays the current date and time.

```

import datetime

# Get the current date and time
current_datetime = datetime.datetime.now()

# Format the date and time (e.g., DD/MM/YYYY HH:MM:SS)
formatted_datetime = current_datetime.strftime("%d/%m/%Y
%H:%M:%S")

# Display the formatted date and time
print("Receipt Date and Time:", formatted_datetime)

```

Task 4: MasterMind

Mastermind is a code-breaking game invented in 1970 by Mordecai Meirowitz, an Israeli postmaster and telecommunications expert.



In this version of MASTERMIND, the computer has the role of **codemaker** and the player that of **codebreaker**. The codemaker randomises a pattern of four coloured pins from six available colours. This colour pattern is hidden from the codebreaker. The colour pattern can contain colour duplicates; for instance, the pattern could be four of the same colour! The available colours are:

BLUE - **GREEN** - **PURPLE** - **RED** - **YELLOW** - **WHITE**

The codebreaker tries to guess the pattern, in both order and colour, within ten turns. With every guess the codemaker provides feedback by displaying the number of correct coloured pins guessed and the number of correct pin positions, as shown in the examples below:

	Example 1				Example 2			
Code	RED	BLUE	PURPLE	BLUE	WHITE	BLUE	YELLOW	RED
User Guess	RED	WHITE	BLUE	BLUE	RED	RED	GREEN	YELLOW
Feedback	Pins Guessed: 3 Correct Pin Positions: 2				Pins Guessed: 2 Correct Pin Positions: 0			

Assessment Criteria

Task 1: Hotel Room Safe Box						
Program Functionality	User-Friendly Interface	Code Efficiency	Proper use of In-line Text (Comments)	Use of Proper Conventions (Camel Case, meaningful variable names etc.)	Name of Folder & Class/es	User Input
Suitable Prompts / Messages displayed	Proper Randomisation (6-digit pin code)	User's input Validation (pin codes, close safe box & exit program)	Other Features (not listed in the task)	Maximum Score: 20 + 2 for every extra feature		
0 – Not Satisfactorily 1- Partly Satisfactorily 2- Entirely Satisfactorily						

Task 2: Lotto System				
Program Functionality	User Friendly Interface	Code Efficiency	Proper use of In-line Text (Comments)	Use of Proper Conventions (Snake Case, meaningful variable names etc.)
Name of Folder & Class/es	User Input	Suitable Prompts / Messages displayed	Numbers Drawn Automatically (5 numbers)	Validation (non-duplicates & valid numbers)
Arithmetic Calculations (total numbers guessed & prize won)	Proper Use of Data Structure (such as Arrays)	Extra Features (not listed in the task)	Maximum Score: 24 + 2 for every extra feature	
0 – Not Satisfactorily 1- Partly Satisfactorily 2- Entirely Satisfactorily				

Task 3: Point of Sale System				
Program Functionality	User Friendly Interface	Code Efficiency	Proper use of In-line Text (Comments)	Use of Proper Conventions (Snake Case, meaningful variable names etc.)
Name of Folder & Class/es	User Input	Suitable Prompts / Messages displayed	Options Validation (Login Screen, Main Menu & New Transaction)	Functionality Validation (Issuing of Receipt)
Ignoring Case Sensitivity (when searching for username & item)	Proper Use of Data Structure (such as Arrays)	Searching of Records (Stock & Cashiers)	Arithmetic Calculations (Subtotal, VAT & Total)	Generating Receipt Number (8-digit Number)
Display Receipt (simulating a real receipt as much as possible)	Display list of items in stock (including formatting)	Extra Features (not listed in the task)	Maximum Score: 34 + 2 for every extra feature	
0 – Not Satisfactorily 1- Partly Satisfactorily 2- Entirely Satisfactorily				

Mastermind				
Program Functionality	User Friendly Interface	Code Efficiency	Use of In-line Text (Comments)	Use of Proper Conventions (Camel Case, Variable names etc.)
Name of Folder & Class/es	User Input	Suitable Messages displayed	Proper Randomisation (Colour Pattern)	Proper Validation of the User's input (Colours chosen)
Ignoring Case Sensitivity (when guessing colours)	Proper Use of Data Structure (such as ARRAYS)	Other Features (not listed in the task)	Maximum Score: 24 + 2 for every extra feature	
- Not Satisfactorily 1- Partly Satisfactorily 2- Entirely Satisfactorily				

Section B

Self-Reflection Exercise

The following ten questions will guide your reflection on the work that you did and the effort that you invested in preparation for Code.Sprint.

Answer all questions:

1. To what extent did you manage to complete the tasks? Rate 1-5.
[1 being all tasks were not successfully finished and 5 being all tasks were successfully finished]
2. What did you learn from completing these tasks?
3. What did you like most about these tasks? Why?
4. What difficulties did you encounter in tackling these tasks? How did you solve them?
5. Which tasks / part of tasks did you find most challenging?
6. What level of support from your mentor did you seek in completing the tasks to the level you did?
7. What other resources have you used in completing this workbook?
8. Mention some testing procedures that you carried out for each task.
9. Suggest further improvements to your work.
10. What did you learn from completing this workbook?

Each question is awarded 10 marks, of which:

- 7 marks are given for the content, and
- 3 marks for using a reflective rather than a descriptive approach.

Final Presentation Guidelines

Prepare a presentation to briefly outline your Code.Sprint journey including:

- An introduction of:
 - yourself,
 - your coding backgrounds,
 - your past Code.Sprint experiences (if any).
- The problem-solving strategies that you employed in the different tasks.
 - If any, identify situations where you have found more than one solution to tackle part/full task and on what grounds did you choose one strategy over another.
- The difficulties you encountered, and which were:
 - solved and how did you overcome your difficulties,
 - not solved.
- The lessons you have learned from this experience.
- Your suggestions to future candidates.
- Your suggestions to Code.Sprint organisers.

Your presentation will be assessed as follows:

40%	30%	15%	10%	5%
Showing a reflective approach, not just demonstrating a descriptive journal of your experience.	Using at least three different sources of support, such as mentor's support and online sources.	Answering all above questions.	Completing your presentation in maximum of five minutes.	Showing a sense of originality in your presentation.

Your presentation:

- can take any format that you prefer such as a slide-show, word-processed document, a video, a website or any other preferred presentation method,
- should not exceed five minutes, should be presented during online a day or two before the Award Ceremony. *

* You will receive instructions via email after the Final Round of the competition.

Contact Us:



www.codesprintmalta.edu.mt



[Code Sprint Malta](#)

*A project created by the Computing Department,
within the Directorate for STEM & Vocational Programmes (DSVP)*

