## SINGAPORE POLYTECHNIC

2024/2025 SEMESTER ONE MID-SEMESTER TEST

COMMON INFOCOMM TECHNOLOGY PROGRAMME

DIPLOMA IN APPLIED AI & ANALYTICS

DIPLOMA IN CYBERSECURITY & DIGITAL FORENSICS

DIPLOMA IN INFORMATION TECHNOLOGY

FIRST YEAR FULL TIME

**FUNDAMENTALS OF PROGRAMMING**

Time Allowed: 2 Hours

Name : \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Class : \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Admission No. : \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## Instructions to Candidates

1. This paper consists of **10** pages (inclusive of cover page).
2. Answer **ALL** questions.

* This is a **RESTRICTED** **OPEN BOOK Test.** You may refer to Lecture notes, Practical solutions on **your laptop ONLY**. **NO** physical notes are allowed.
* You are **NOT** allowed to access the module materials on the BrightSpace and online material related to JavaScript coding.
* **Do NOT use any communication software or access any websites that provide online e-communication (e.g. Web WhatsApp ,etc). Any form of communication online or offline is prohibited. Anyone caught doing so will be considered cheating which may result in FAILING the module or all the modules, suspension or expulsion.**
* **You are NOT allowed to use Generative AI tool (example: ChatGPT co-pilot, Visual Studio Code Co-Pilot or any form of Generative AI plug-in) for the Test. Anyone caught doing so will be considered cheating which may result in failing the module or all the modules, suspension or expulsion.**

1. You are **NOT** allowed to use any built-in sorting or array functions or sorting libraries in your solution. The construct forEach() is **NOT** allowed.
2. Only loops syntax taught in lecture slides are accepted.
3. Ask from invigilator for A4 paper if you need for rough work.
4. You are **NOT** allowed to connect to wi-fi during the test unless instructed by the invigilator.
5. You are required to **return the Question paper** before you leave the room**.**

**Instructions BEFORE you start the test:**

1. Turn on Wi-Fi. Login to the course site for ST0523 FOP in BrightSpace.
2. Check that your question paper is Set A (refer to the top right corner of this paper).
3. Go to BrightSpace unit **Continual Assessment > MST to** download **SETA.js**
4. Make a copy of SETA.js in your FOP folder or your own folder where you usually run JavaScript program.
5. Rename the file using the following convention i.e. **AxxxxxName.js** where xxxxxx is your admin number followed by your name. The file extension must be .js

***For example :***

*The file A12345TanAhMeng.js is done by Tan Ah Meng, admin number 12345.*

1. You should only use this file you have renamed to do the practical test and for submission.
2. Once you are ready, **TURN OFF** your wi-fi.
3. Do **NOT** start until the invigilator tells you to do so.

**Instructions when you have completed the test or when the duration of test is up:**

1. Ensure that your JavaScript file is renamed according to the instructions above.
2. Stop coding once the test is over, the invigilator will inform you.
3. Wait for the invigilator’s instructions to allow you to turn on the wi-fi on your laptop.
4. You are only given **5minutes** to submit your file in BrightSpace.

* submit your program e.g. A12345TanAhMeng to this folder :

**Continual Assessments > MST > MST (2024) Submission**

* only ONE submission is allowed

1. Do **NOT** delete your program from your laptop and do **NOT** modify it. An exact copy MUST be kept in your laptop after you have submitted a copy to BrightSpace.
2. Return the Question paper to the invigilator(s) before you leave the venue.

## **JavaScript Syntax List**

**Operator Precedence**

|  |
| --- |
| ++var ,var++, --var, var--  ! (Not)  \*\* (Exponentiation)  \*, /, % (Multiplication, Division and Modulus)  +, - (Binary addition and subtraction)  <, <=, >, >= (Comparison)  ==, !=; (Equality)  && (Conditional AND)  || (Conditional OR)  =, +=, -=, \*=, /=, %= (Assignment) operator)  **Highest**  **Lowest** |

|  |  |  |
| --- | --- | --- |
| **The if statement**  if ( *condition* ) {  *// block of code*  } | **The else statement**  if ( *condition* ) {  *// block of code*  } else {  *// block of code*  } | **The else if statement**  if ( *condition1* ) {  *// block of code*  } else if ( *condition2* ) {  *// block of code*  } else {  *// block of code*  } |
| **The for loop**  for (*statement 1; statement 2; statement 3)* {  *// block of code*  } | | **The while loop**  while ( *condition )* {  *// block of code*  } |
| Math.floor(Math.random( ) \* x ) will return a random integer between  0 and (x-1).  Math.abs(x) returns the absolute value of a number.  e.g Math.abs(-4) results in 4 | | |
| process.stdout.write(); **//Continuously prints information and**  **does not add new line.**  console.log(“xxxx” ); **//prints content xxxx on a new line** | | |
| **//sample codes to prompt user to enter an input**  const input = require('readline-sync');  let userInput = input.question("Enter name: ");  console.log("My name is " + userInput); | | |
| To install readline sync at command prompt>npm install readline-sync | | |
| Common command prompts :   |  |  | | --- | --- | | **Command Prompt** | **Remarks** | | dir | To list the files/folders in the current folder, assuming FOP has 3 folders (P1, P2 and P3) and 1 file (f1.js):  **C:\FOP > dir** | | cd.. | To navigate one level up :  **C:\FOP> cd..**  C:\> | | cd P1 | To navigate to folder P1:  **C:\FOP> cd P1**  C:\FOP\P1> | | mkdir P4 | To make a new folder P4:  **C:\FOP>mkdir P4** | | rmdir P3 | To remove folder P3:  **C:\FOP>rmdir P3**    Folder P3 is deleted | | | |

You are an IT staff of a local cinema called ‘Starlight Cinema’. You are tasked to develop a JavaScript program to manage the cinema's operations and provide some simple functionalities. You are to **follow** the text to be displayed in **ALL** output. The following table shows the cinema information.

|  |  |  |
| --- | --- | --- |
| **No.** | **Cinema Information** | **Sample Data** |
| 1. | Address | 45 Orchard |
| 2. | Cinema rating | 8.5 |
| 3. | Average daily revenue ($) | 45000.00 |
| 4. | Max daily screenings | 12 |
| 5. | Does cinema support 3D movies? | true or false |
| 6. | Seating Capacity | 350 |

Use the JavaScript file you have renamed and start coding the following requirements. You are **NOT** supposed to change the codes that are given in the file.

1. Write 3 **comment lines** at the top of the program to provide your details, for example :

Name : Tan Meng Heng *(your name)*

Class : DIT/FT/1A/993 *(your class)*

Admin no : 240123 *(your admin number)*

(2 marks)

1. In the JavaScript program, fill in the most appropriate primitive data types for each cinema information. The data type for **Cinema Name** is shown as an example. Do **NOT** write here (it will **NOT** be marked), type in your JavaScript program.

|  |
| --- |
| // **Fill in the blanks with the most appropriate data types**  /\* **Example:  Cinema Name**  Data Type: string   1. **Address**   Data Type: \_\_\_\_\_\_\_\_\_\_\_   1. **Cinema rating**   Data Type: \_\_\_\_\_\_\_\_\_\_\_   1. **Average daily revenue**   Data Type: \_\_\_\_\_\_\_\_\_\_\_   1. **Max daily screenings**   Data Type: \_\_\_\_\_\_\_\_\_\_\_   1. **Support 3D Movies?**   Data Type: \_\_\_\_\_\_\_\_\_\_\_   1. **Seating Capacity**   Data Type: \_\_\_\_\_\_\_\_\_\_\_  \*/ |

(18 marks)

3. Write a function **displayCinema** which:

* accepts two parameters i.e. cinema’s address and maximum daily screenings,
* does not return any value, and
* displays the information in the function. Do **NOT** hardcode.

Invoke the function to display the output as shown below:

|  |
| --- |
| Starlight,located at 45 Orchard has 12 maximum screenings daily. |

(10 marks)

4. Write a function **displayMovie** which:

* accepts one parameter i.e. array **movieInfo,**
* does not return any value, and
* displays all movie names in the function using a ***for*** loop. Do **NOT** hardcode.

Invoke the function to display the output as shown below :

|  |
| --- |
| No. Movie Title  1. ABBA  2. War  3. Doom  4. Joker  5. Cheers |

(15 marks)

5. Write a function **displayRating** which:

* prompts the user to enter an integer, and
* displays the output of corresponding movie and rating from the **movieInfo** array and **ratingInfo** array respectively. Do **NOT** hardcode.

You may assume that the user will always enter a valid integer. **No** data validation is required.

Invoke the function to display the output as shown below :

|  |
| --- |
| Enter a movie number (1-5) to check the rating : **3**  Doom has a rating of 7.4 |

(10 marks)

6. Write a function **displayConvertDuration** which:

* accepts two parameters i.e. **movieInfo** and **durationInfo** arrays
* prompts the user for a movie name
* retrieves the duration of the required movie from the array **durationInfo,**
* convert the duration to hours and minutes format, and
* returns the converted duration to the main program for display.

You may assume that the user will always enter a correct movie listed in **movieInfo** array.

**No** data validation is required. Note: Zero hour or minute should NOT be printed (refer to sample output 2).

Invoke the function to display the output as shown below, do **NOT** hardcode:

Sample output 1 displayed in the main program, text in **bold** is the user input :

|  |
| --- |
| Enter a movie name for the duration: **Joker**  Joker is 2hr(s) and 30min(s) |

Sample output 2 displayed in the main program, text in **bold** is the user input :

|  |
| --- |
| Enter a movie name for the duration: **Cheers**  Cheers is 2hr(s) |

(15 marks)

7. Write a function **compareRating** which :

* accepts two parameters i.e. arrays **ratingInfo** and **movieInfo,**
* finds the lowest rating from all the ratings
* calculate the average rating of all the movies
* calculate the difference between the lowest rating and the average rating, and
* returns a string to be displayed in main program as shown below.

|  |
| --- |
| Movie War has the lowest rating of 6.3  Average rating is 7.9  Difference is 1.6 |

(15 marks)

8. Write a function **displayMenu** which:

* does not take in any parameters nor return any value, and
* displays a menu for users’ selection as shown below.

|  |
| --- |
| Menu:  1. Cinema Details  2. Movie Titles  3. Display Rating  4. Convert Duration  5. Compare Rating  0. Exit  Enter an option 1-5, or 0 to exit: |

In the main program, invoke **displayMenu** to prompt for user input. The respective function will be invoked based on the user’s input.

|  |  |  |
| --- | --- | --- |
| **User Input** | **Menu options** | **Function to invoke/Remarks** |
| 1 | Cinema Details | displayCinema |
| 2 | Movie Titles | displayMovie |
| 3 | Display Rating | displayRating |
| 4 | Convert Duration | displayConvertDuration |
| 5 | Compare Rating | compareRating |
| 0 | Exit | Terminates the program |

**Validation of user input is required** ie your program can only accept any values from 0 to 5.

If 0 is entered, the program will terminate. You are **NOT** allowed to use **for** loop.

Sample output if any invalid input is entered e.g. **-5.** It displays an error message and re-prompts user for an input. Text in **bold** are user input.

Menu:

1. Cinema Details

2. Movie Titles

3. Display Rating

4. Convert Duration

5. Compare Rating

0. Exit

Enter an option 1-5, or 0 to exit: **-5**

Invalid choice, please re-enter 0 to 5 only

Menu:

1. Cinema Details

2. Movie Titles

3. Display Rating

4. Convert Duration

5. Compare Rating

0. Exit

Enter an option 1-5, or 0 to exit: **1**

Starlight,located at 45 Orchard has 12 maximum screenings daily.

Sample output when 0 is entered, it displays a message and terminates the program. Text in **bold** is the user input.

|  |
| --- |
| Menu:  1. Cinema Details  2. Movie Titles  3. Display Rating  4. Convert Duration  5. Compare Rating  0. Exit  Enter an option 1-5, or 0 to exit: **0**  Goodbye |

(15 marks)

**- End of Paper –**