

---

# **Creative Integrated Coding**

---

**2019**

**IQCS**

# First written test

Creative Integrated Coding

| Field                                  | Coding area  | Qualification   | Creative Integrated Coding            | Valid date   | 2019         |
|--|--|---|---------------------------------------|--------------|--------------|
| Exam type                              | Multiple choice/<br>Short answer                             | No. of Ques.  | 1 <sup>st</sup> 40 multiple questions | Exam hour    | 90 minutes   |
| Title                                  | Sub title  | Detail  |                                       | No. of Ques. | Distribution |
| Computing thinking and problem solving | 1. Understanding and applying computing thinking             | <ul style="list-style-type: none"> <li>• Understanding and expressing the concepts of data and information</li> <li>• Distinguishing and utilizing the type of information</li> <li>• Expressing various types of information digitally</li> <li>• Understanding and utilizing the components of computing thinking.</li> </ul>   |                                       | 12           | 30           |
|  | 2. Problem analysis and structuralization                    | <ul style="list-style-type: none"> <li>• Understanding and analyzing a given problem</li> <li>• Organizing and expressing data in various ways</li> <li>• Understanding and structuring the concepts of linear and nonlinear structures</li> </ul>  |                                       |              |              |
|  | 3. Solving problems in daily life through computing thinking | <ul style="list-style-type: none"> <li>• Simplifying a given problem</li> <li>• Understanding and applying abstraction</li> <li>• Finding and formulating a pattern by searching for certain trends and rules that are repeated</li> <li>• Explaining how to solve the problem in order</li> <li>• Finding various ways to solve problems and choosing the appropriate ones</li> <li>• Explaining the problem of problem-solving and how to improve.</li> </ul> |                                       |              |              |
| Algorithm design                       | 1. Creating Algorithms for problem-solving                   | <ul style="list-style-type: none"> <li>• Understanding algorithms</li> <li>• Designing algorithms</li> <li>• Expressing algorithm</li> <li>• Finding and fixing errors in the algorithm</li> <li>• Modifying the algorithm with more effectively</li> </ul>   |                                       | 12           | 30           |
|  | 2. Algorithm Design of Complex Structures                    | <ul style="list-style-type: none"> <li>• Understanding the relationship between computer function and algorithm</li> <li>• Predicting the results of the operation of the algorithm</li> <li>• Analyzing algorithms</li> <li>• Understanding the control structure of algorithms</li> <li>• Expressing the control structure of algorithms in a complex way</li> </ul>  |                                       |              |              |

| Field  | Coding area                               | Qualification   | Creative Integrated Coding            | Valid date   | 2019         |
|--|---|---|---------------------------------------|--------------|--------------|
| Exam type  | Multiple choice/<br>Short answer          | No. of Ques.  | 1 <sup>st</sup> 40 multiple questions | Exam hour    | 90 minutes   |
| Title  | Sub title                                 | Detail  |                                       | No. of Ques. | Distribution |
| Understanding the programming language and programming | 1. Understanding the programming language | <ul style="list-style-type: none"> <li>Knowing the beginning and end of the program</li> <li>Explaining the implementation procedure of the programming language</li> <li>Understanding and using conditional statements and loop statements</li> <li>Understanding and using variables and operators</li> </ul>  |                                       | 16           | 40           |
|  | 2. Programming design                     | <ul style="list-style-type: none"> <li>Understanding the problem conditions and requirements</li> <li>Designing an efficient program</li> <li>Checking for program errors and correct them</li> <li>Understanding and programming complex structures</li> </ul>   |                                       |              |              |
|  | 3. Block programming                      | <ul style="list-style-type: none"> <li>Knowing the screen composition and key terms</li> <li>Using sequential, iterative structures for a given situation</li> <li>Writing a program using multiple choice, multiple iterations</li> <li>Making a program that makes different movements considering various conditions</li> <li>Understanding variables and constants, and using them to write input/output program</li> <li>Understanding and using the coordinates to write a program</li> <li>Writing a program knowing the difference between signal and replication</li> <li>Organizing more than one scene through a scene connection</li> <li>Writing a program using a function</li> <li>Writing a program using the list</li> </ul> |                                       |              |              |

## 2<sup>nd</sup> practical test

### Creative Integrated Coding Level I

| Field                | Coding area                            | Qualification  | Creative Integrated Coding | Valid date   | 2019         |
|----------------------|--|--|----------------------------|--------------|--------------|
| Exam type            | Block Coding                           | No. of Ques.   |                            | Exam hour    | 120 minutes  |
| Title                | Sub title                              | Detail   |                            | No. of Ques. | Distribution |
| Understanding Coding | 1. Problem - solving                   | <ul style="list-style-type: none"> <li>Expressing the problem resolution in pictures or symbols</li> <li>Expressing the problem-solving process in a flowchart</li> </ul>  |                            |              |              |
|                      | 2. Algorithm                           | <ul style="list-style-type: none"> <li>Understanding that algorithms are the order in which things happened</li> <li>Expressing the repetition structure in a flowchart</li> <li>Expressing the selection structure as a flowchart</li> <li>Finding and correcting errors in algorithms</li> <li>Finding the most effective algorithm</li> </ul> |                            |              |              |
| Practice of coding   | 1. Block Coding (sequence, repetition) | <ul style="list-style-type: none"> <li>Using block commands in order to resolve the 'Go to school' mission</li> <li>Solving the mission efficiently through the '~ Repeat' command</li> </ul>  |                            | 2            | 100          |
|                      | 2. Block Coding (sequence, parallel)   | <ul style="list-style-type: none"> <li>Knowing that each of the blocks assembled on each sprite is executed at the same time and using the 'wait' block to implement the conversation</li> <li>Creating conversation animations of different sprites in consideration of the order of time</li> </ul>  |                            |              |              |
|                      | 3. Block Coding (Selection)            | <ul style="list-style-type: none"> <li>Giving different commands depending on the situation through the 'if~' command.</li> <li>Giving different commands depending on the situation through the 'if ~ or' command</li> </ul>  |                            |              |              |
|                      | 4. Block Coding (signal)               | <ul style="list-style-type: none"> <li>Making a signal and create a work that sends and receives a promised signal between sprites</li> </ul>  |                            |              |              |
|                      | 5. Block Coding (variable)             | <ul style="list-style-type: none"> <li>Knowing the need for variables and controlling your data</li> </ul>   |                            |              |              |

## 2<sup>nd</sup> practical test

Creative Integrated Coding Level II

| Field                | Coding area   | Qualification  | Creative Integrated Coding | Valid date   | 2019         |
|----------------------|---|--|----------------------------|--------------|--------------|
| Exam type            | Block Coding  | No. of Ques.   |                            | Exam hour    | 120 minutes  |
| Title                | Sub title   | Detail   |                            | No. of Ques. | Distribution |
| Understanding Coding | 1. Problem-solving  | <ul style="list-style-type: none"> <li>Expressing the problem resolution in pictures or symbols</li> <li>Expressing the problem-solving process in a flowchart</li> </ul>  |                            |              |              |
|                      | 2. Algorithm  | <ul style="list-style-type: none"> <li>Understanding that algorithms are the order in which things happened</li> <li>Expressing the repetition structure in a flowchart</li> <li>Expressing the selection structure as a flowchart</li> <li>Finding and correcting errors in algorithms</li> <li>Finding the most effective algorithm</li> </ul> |                            |              |              |
| Practice of coding   | 1. Creating a Game<br>Block coding (sequence, repetition, condition, signal, variable, operand) | <ul style="list-style-type: none"> <li>Making the score increase by using variables</li> <li>Using the stopwatch to make a 30-second game</li> </ul>   |                            | 2            | 100          |
|                      | 2. Math coding (Create a arithmetic operation calculator)                                       | <ul style="list-style-type: none"> <li>Creating a flow charts to match the algorithm and create a script accordingly</li> <li>Calculating the input data and output the correct result</li> </ul>  |                            |              |              |
|                      | 3. Math coding (expressing an analogue clock)   | <ul style="list-style-type: none"> <li>Creating a flow charts to match the algorithm and create a script accordingly</li> <li>Understanding and accurately controlling angles</li> </ul>   |                            |              |              |
|                      | 4. Math coding (expressing an analogue clock)   | <ul style="list-style-type: none"> <li>Understanding and using comparative calculations and temporary variables</li> <li>Understanding the arrangement and using the list to deal with the data</li> </ul>   |                            |              |              |
|                      | 5. Math coding (finding symmetric number)   | <ul style="list-style-type: none"> <li>Understanding mathematical concepts, implementing algorithms that match them on your own and writing scripts</li> </ul>   |                            |              |              |
|                      | 6. Math coding (finding the greatest common divisor, least common multiple)                     | <ul style="list-style-type: none"> <li>Performing the improvement process for the simple algorithm implementation</li> <li>Modifying the script to reduce execution time</li> </ul>  |                            |              |              |

## 2<sup>nd</sup> practical test

Creative Integrated Coding Level III

| Field                                   | Coding area   | Qualification  | Creative Integrated Coding | Valid date          | 2019                |
|---|---|--|----------------------------|---------------------|---------------------|
| <b>Exam type</b>                        | Block Coding  | <b>No. of Ques.</b>  |                            | <b>Exam hour</b>    | <b>120 minutes</b>  |
| <b>Title</b>                            | <b>Sub title</b>                                    | <b>Detail</b>  |                            | <b>No. of Ques.</b> | <b>Distribution</b> |
| Algorithm and programming               | 1. Information of representation and management     | • Information representation   |                            | 2                   | 100                 |
|   |   | • Information management   |                            |                     |                     |
|   | 2. Practice of computing thinking                   | • Structuralization of problem   |                            |                     |                     |
|   |   | • Abstraction of problem   |                            |                     |                     |
|   |   | • Modeling and simulation  |                            |                     |                     |
|   | 3. Practic of algorithm                             | • Algorithm design of complex structure  |                            |                     |                     |
| • Analysis and evaluation of algorithms |   |  |                            |                     |                     |
| Project execution                       | 1. Creative problem-solving                         | • Developing an algorithm for problem solving by discovering creative ideas related to real life |                            |                     |                     |
|   | 2. Physical computing                               | • Realizing the product by appropriately using sensors and actuators                             |                            |                     |                     |
|   | 3. Debugging  | • Finding errors in programming and circuit configurations and suggesting ways to solve them     |                            |                     |                     |
| Computing and problem-solving           | 1. Integrating activity based on computing thinking | • Programming and integrating  |                            |                     |                     |
|   |   | • Team project creation and evaluation   |                            |                     |                     |