Xi'an Jiaotong-Liverpool University

西交利物浦大學

Paper CODE	EXAMINER	EMAIL OF EXAMINER	TEL
CPT105/2023/24 S1 Final Exam			

1st SEMESTER 2023/24 FINAL EXAMINATION

Undergraduate – Year 1

INTRODUCTION TO PROGRAMMING IN JAVA

TIME ALLOWED: 2 HOURS

INSTRUCTIONS TO CANDIDATES

1. **This is a Closed-book examination. Please tick the integrity disclaimer when handing in your answer sheets. Please complete the assessment independently and honestly.**

2. **The total marks are 100.**

3. **The exam paper consists of 11 printed pages.**

4. **This exam consists of 2 sections:**

   Section A consists of single-choice questions with 20 questions worth for a total mark of 60.

   Section B consists of coding tasks with two questions. Each question is worth 20 marks.

   Answer all questions. There is NO penalty for providing a wrong answer.

5. **Only English solutions are accepted. Answers need to be handwritten on answer sheets. Please clearly indicate the question numbers before your solutions. Please also indicate ONLY your student ID number on answer sheets.**

6. **The exam duration is 2 hours. If any major problems are preventing you from continuing the exam or submitting your answers in time, please do not hesitate to email the Module Examiner or Assessment Team of Registry (assessment@xjtlu.edu.cn).**

Section A. removed.

Section B. Removed.

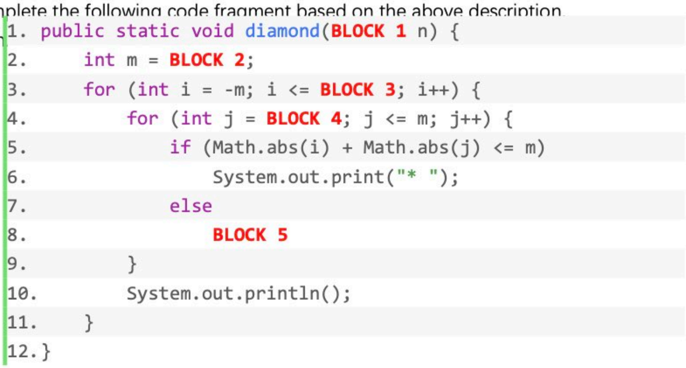
Section C. (2 * 20 = 40 mark)

(20 mark) A Diamond is FORever. Here is a Java method *void diamond(int n)* that takes an odd integer argument n and prints an n-by-n pattern like the test cases below, with an asterisk (*) for each element which makes up a diamond, and a dot (.) for each entry that is not, and one space between each * or .

Test case:
diamond(5)

```
. . * . .

. * * * .

* * * * *

. * * * .

. . * . .
```

Please complete the following code fragment based on the above description.
Note: Math



```java
public static void diamond(BLOCK 1 n) {
    int m = BLOCK 2;
    for (int i = -m; i <= BLOCK 3; i++) {
        for (int j = BLOCK 4; j <= m; j++) {
            if (Math.abs(i) + Math.abs(j) <= m)
                System.out.print("* ");
            else
                BLOCK 5
        }
        System.out.println();
    }
}
```

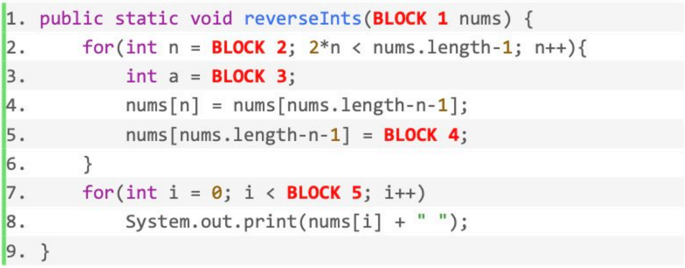
(4 mark) Block 1:

(4 mark) Block 2:
(4 mark) Block 3:
(4 mark) Block 4:
(4 mark) Block 5:

(20 mark) Reverse Integers. Here is a Java method *void reverseInts(int[] nums)* that reverse the values within the input array.

Test cases:
reverseInts(new int[]{1, 2, 3, 4, 5}) → 5 4 3 2 1
reverseInts(new int[]{1, 2, 3, 4}) → 4 3 2 1

Please complete the following code fragment based on the above description.



```
public static void reverseInts(BLOCK 1 nums) {
    for(int n = BLOCK 2; 2*n < nums.length-1; n++){
        int a = BLOCK 3;
        nums[n] = nums[nums.length-n-1];
        nums[nums.length-n-1] = BLOCK 4;
    }
    for(int i = 0; i < BLOCK 5; i++)
        System.out.print(nums[i] + " ");
}
```

(4 mark) Block 1:
(4 mark) Block 2:
(4 mark) Block 3:
(4 mark) Block 4:
(4 mark) Block 5: