

NATIONAL UNIVERSITY OF SINGAPORE
CS1101S — PROGRAMMING METHODOLOGY

1

(AY2019/2020 SEMESTER 1)

READING ASSESSMENT 1

Time Allowed: **45 Minutes**

INSTRUCTIONS

1. This question paper comprises **EIGHT (8)** printed pages, including this page.
2. You are also provided with **one OCR Form** to write your answers.
3. Clearly **write** and **shade** your **STUDENT NUMBER** on your **OCR Form** using a **2B PENCIL**.
4. There are **15** multiple-choice questions. Each question has one correct answer. **1 mark** is awarded for each correct answer and there is no penalty for a wrong answer.
5. The full score is **15 marks**.
6. Answer **ALL** questions.
7. Use only a **2B PENCIL** to **shade** your answers on your **OCR Form**.
8. This is a **CLOSED BOOK** assessment, but you are allowed to bring in one A4 sheet of notes (handwritten or printed on both sides).
9. **Submit only the OCR Form.** (We will also collect this question paper back from you.)

- (1) What is the single-digit **number** at the **top-right corner** on the **front page** of this question paper? (**Important:** Please make sure your answer is correct because it determines how we mark your answers to all the subsequent questions.)

- A. 1
- B. 2
- C. 3
- D. 4
- E. 5

Scoping

- (2) What is the result of evaluating the following Source program?

```
const x = 2;
const y = 5;
const z = 10;

function f(z) {
    const y = 3;
    return x + y + z;
}
f(2 * z);
```

- A. 15
- B. 17
- C. 25
- D. 27
- E. Error: one or more names is/are redeclared

- (3) What is the result of evaluating the following Source program?

```
const y = 7;

function f(y) {
    const w = 6;

    function g(x) {
        const w = 8;
        return x + y + 20;
    }
    return g(10 + w);
}
f(5);
```

- A. 39
- B. 41
- C. 43
- D. 45
- E. Error: one or more names is/are redeclared

- (4) What is the result of evaluating the following Source program?

```
function g(x) {
  function g(x) {
    function g(x) {
      return (x <= 3) ? 34 : g(x - 3);
    }
    return (x <= 2) ? 23 : g(x - 2);
  }
  return (x <= 1) ? 12 : g(x - 1);
}
g(100);
```

- A. 12
- B. 23
- C. 34
- D. Error: name g is redeclared
- E. None of the above

- (5) What is the result of evaluating the following Source program?

```
function f(x, y) {
  return (y, x) => 3 * x + y;
}
f(1, 2)(3, 4);
```

- A. 5
- B. 7
- C. 13
- D. 15
- E. None of the above

- (6) What is the result of evaluating the following Source program?

```
function g(f) {
  return x => f(x) + f(f(x));
}
g(x => 2 * x + 1)(3);
```

- A. 56
- B. 22
- C. 14
- D. 7
- E. None of the above

- (7) Given the following function declarations, which of these expressions produces the same result as `f(7, 8, 9)`?

```
function f(x, y, z) {
    return x + 2 * y + 3 * z;
}
function g() {
    return z => (y => (x => x + 2 * y + 3 * z));
}
```

- A. `g(7)(8, 9)`
- B. `g(9)(8)(7)`
- C. `g(7)(8)(9)`
- D. `g()(7)(8)(9)`
- E. `g()(9)(8)(7)`

Processes

In some of the following questions, the pre-declared `display` function is used in the Source programs. The `display` function displays/prints the value of its input argument in the console. For example, `display(2 * 5)` prints `10`, and `display(1 > 2)` prints `false`.

- (8) What kind of process does the following function `f` give rise to for any *odd integer* argument `n > 1`?

```
function f(n) {
    return (n <= 1)
        ? n
        : (n % 2 === 0)
            ? f(n - 1) * 2
            : f(n - 2);
}
```

- A. An iterative process
- B. A recursive process
- C. A process that is both iterative and recursive
- D. A substitution process
- E. An alternating process

- (9) What is the sequence of values printed by the `display` function when the following program is evaluated?

```
function f(n) {
    display(n);
    return (n <= 1)
        ? 1
        : (n === 2) ? 2 : n * f(n - 1);
}
f(6);
```

- A. 2 3 4 5 6
- B. 1 2 3 4 5 6
- C. 6 5 4 3 2 1
- D. 6 5 4 3 2
- E. 6 5 4 3

- (10) What kind of process does the following function `f` give rise to for any *integer* argument $n > 1$?

```
const d = (x) => 2 * x;
function f(n) {
    return (n <= 1) ? 2 : d(f(n - 1));
}
```

- A. An iterative process
- B. A recursive process
- C. A process that is both iterative and recursive
- D. A substitution process
- E. An abstraction process

- (11) What is the sequence of values printed by the `display` function when the following program is evaluated?

```
function d(x) {
    display(2 * x);
    return 2 * x;
}
function f(n) {
    display(n);
    return (n <= 1) ? 2 : d(f(n - 1));
}
f(5);
```

- A. 5 4 3 2 1 4 8 16 32
- B. 5 10 4 8 3 6 2 4 1
- C. 5 4 4 8 3 16 2 32 1
- D. 5 2 4 8 16 32 4 3 2 1
- E. None of the above

- (12) What kind of process does the following function `f` give rise to for any *integer* argument `n > 1`?

```
function f(n) {
    return n === 1 ? 1 : f(f(n - 1));
}
```

- A. An iterative process
- B. A recursive process
- C. A process that is both iterative and recursive
- D. A process that is neither iterative nor recursive
- E. An infinite process

- (13) What is the sequence of values printed by the `display` function when the following program is evaluated?

```
function fib(n) {
    display(n);
    return (n <= 1) ? n : fib(n - 1) + fib(n - 2);
}
fib(4);
```

- A. 4 3 2 2
- B. 4 3 2 1 1 2 1
- C. 4 2 3 0 1 1 2 0 1
- D. 4 3 2 2 1 1 0 1 0
- E. 4 3 2 1 0 1 2 1 0

Correctness

- (14) We specify that the function *S*, when applied to two *positive integer* arguments, should return the sum of the arguments. Consider the following implementation:

```
function S(a, b) {
    return a + b;
}
```

Which one of the following statements is correct?

- A. The function *S* meets the specification.
- B. The function *S* always returns an incorrect result for any two positive integer arguments.
- C. The function *S* does not meet the specification because it can be applied to non-positive and/or non-integer arguments.
- D. The function *S* does not meet the specification because it can be applied to arguments that are strings.
- E. The function *S* does not meet the specification because it does not check whether the arguments are valid inputs.

- (15) We specify that the function F , when applied to any number argument, should return the *floor* of the argument. The *floor* of a number x is the largest integer that is less than or equal to x . Consider the following implementation:

```
function F(x) {
  function h(k) {
    return (k > x) ? k - 1 : h(k + 1);
  }
  return h(1);
}
```

Which one of the following statements is correct?

- A. The function F meets the specification.
- B. The function F returns the correct result only when the number argument is an integer.
- C. The function F returns the correct result only when the number argument is non-negative.
- D. The function F returns the correct result only when the number argument is positive.
- E. The function F does not meet the specification because it is slow when the number argument is large.

————— END OF QUESTIONS —————

(blank page)