

NATIONAL UNIVERSITY OF SINGAPORE
CS1101S — PROGRAMMING METHODOLOGY

(AY2021/2022 SEMESTER 1)

READING ASSESSMENT 2

Time Allowed: **45 Minutes**

INSTRUCTIONS

1. This assessment contains **22 Multiple-Choice Questions** in **4 Sections**.
2. Each question has one correct answer. **1 mark** is awarded for each correct answer and there is no penalty for a wrong answer.
3. The full score of this assessment is **22 marks**.
4. Answer **all questions**.
5. This is a **Closed-Book** assessment, but you are allowed one double-sided **A4 / foolscap / letter-sized sheet** of handwritten or printed **notes**.
6. You are allowed to use up to **4 sheets** of **blank A4 / foolscap / letter-sized** paper as **scratch paper**.
7. You are allowed access to these online reference pages:
 - **Source §3 pre-declared constants and functions** at https://docs.sourceacademy.org/source_3/global.html
 - **Specification of Source §3** at https://docs.sourceacademy.org/source_3.pdf
8. **Follow the instructions of your invigilator or the module coordinator to submit your answers.**

Section A

For all the questions in this section, consider the following Source program:

Program A:

```
function foo(xs, ys) {  
  if (is_null(xs)) {  
    return map(x => x, ys);  
  } else {  
    const wish = foo(tail(xs), ys);  
    set_tail(xs, wish);  
    return xs;  
  }  
}  
  
const AA = list(list(1), list(2));  
const BB = list(list(3), list(4));  
const CC = foo(AA, BB);  
  
const AA2 = tail(tail(AA));  
const BB2 = tail(tail(BB));  
const CC2 = tail(tail(CC));
```

(1) What is the value of AA (in *list notation*) at the end of the evaluation of Program A?

- A. `list(1, 2)`
- B. `list(3, 4)`
- C. `list(1, 2, 3, 4)`
- D. `list(3, 4, 1, 2)`
- E. `list(list(1), list(2))`
- F. `list(list(3), list(4))`
- G. `list(list(1), list(2), list(3), list(4))`
- H. `list(list(3), list(4), list(1), list(2))`
- I. None of the other options is the correct answer

(2) What is the value of BB (in *list notation*) at the end of the evaluation of Program A?

- A. `list(1, 2)`
- B. `list(3, 4)`
- C. `list(1, 2, 3, 4)`
- D. `list(3, 4, 1, 2)`
- E. `list(list(1), list(2))`
- F. `list(list(3), list(4))`
- G. `list(list(1), list(2), list(3), list(4))`
- H. `list(list(3), list(4), list(1), list(2))`
- I. None of the other options is the correct answer

(3) What is the value of CC (in *list notation*) at the end of the evaluation of Program A?

- A. `list(1, 2)`
- B. `list(3, 4)`
- C. `list(1, 2, 3, 4)`
- D. `list(3, 4, 1, 2)`
- E. `list(list(1), list(2))`
- F. `list(list(3), list(4))`
- G. `list(list(1), list(2), list(3), list(4))`
- H. `list(list(3), list(4), list(1), list(2))`
- I. None of the other options is the correct answer

(4) What is the result of the following statement if it is evaluated at the end of Program A?

```
[ CC === AA, CC2 === BB, CC2 === AA2 ];
```

- A. `[false, false, false]`
- B. `[false, false, true]`
- C. `[false, true, false]`
- D. `[false, true, true]`
- E. `[true, false, false]`
- F. `[true, false, true]`
- G. `[true, true, false]`
- H. `[true, true, true]`

(5) What is the result of the following statement if it is evaluated at the end of Program A?

```
[ head(CC) === head(AA),  
  head(CC2) === head(BB),  
  head(CC2) === head(AA2) ];
```

- A. [false, false, false]
- B. [false, false, true]
- C. [false, true, false]
- D. [false, true, true]
- E. [true, false, false]
- F. [true, false, true]
- G. [true, true, false]
- H. [true, true, true]

Section B

For all the questions in this section, consider the following Source program:

Program B:

```
let z = 1;
let y = 1;
let x = 1;

function goo(f) {
  z = 2;
  let y = 2;
  let x = 2;
  let g = () => 100 * x + f(y);
  z = 3;
  y = 3;
  x = 3;
  return g;
}

const hoo = goo(y => 10 * y + z);
z = 4;
y = 4;
x = 4;
const w = hoo();
w;
```

(6) What is the result of evaluating Program B?

- A. 111
- B. 211
- C. 221
- D. 331
- E. 333
- F. 334
- G. 344
- H. 444
- I. None of the other options is the correct answer

(7) How many bindings does the program environment frame contain?

- A. 0
- B. 1
- C. 2
- D. 3
- E. 4
- F. 5
- G. 6
- H. 7
- I. More than 7

(8) How many environment frames get created during the evaluation of Program B? (Do not count the global environment frame.)

- A. 0
- B. 1
- C. 2
- D. 3
- E. 4
- F. 5
- G. 6
- H. 7
- I. More than 7

(9) Of the environment frames that get created during the evaluation of Program B, how many extend the program environment *directly*? (Do not count the global environment frame.)

- A. 0
- B. 1
- C. 2
- D. 3
- E. 4
- F. 5
- G. 6
- H. 7
- I. More than 7

(10) How many function objects get created during the evaluation of Program B? (Do not count function objects of primitive and pre-declared functions such as `array_length`, `math_floor`, `pair`, `head`, `tail`, `length`, and `map`.)

- A.** 0
- B.** 1
- C.** 2
- D.** 3
- E.** 4
- F.** 5
- G.** 6
- H.** 7
- I.** More than 7

(11) Of the function objects that get created during the evaluation of Program B, how many have the program environment as their environment (i.e. their right circles point to the program frame)? (Do not count function objects of primitive and pre-declared functions such as `array_length`, `math_floor`, `pair`, `head`, `tail`, `length`, and `map`.)

- A.** 0
- B.** 1
- C.** 2
- D.** 3
- E.** 4
- F.** 5
- G.** 6
- H.** 7
- I.** More than 7

Section C

For all the questions in this section, consider the following Source program:

Program C:

```
function moo(f, xs) {  
  function dee(g, ys) {  
    if (is_null(ys)) {  
      return ys;  
    } else {  
      return pair(g(head(ys)), moo(g, tail(ys)));  
    }  
  }  
  return dee(f, xs);  
}  
  
let L = list(1, 2, 3);  
let R = moo(w => 10 * w, L);  
R;
```

(12) How many *pairs* get created during the evaluation of Program C?

- A. 0
- B. 1
- C. 2
- D. 3
- E. 4
- F. 5
- G. 6
- H. 7
- I. More than 7

(13) How many environment frames get created during the evaluation of Program C? (Do not count the global environment frame. We assume that the applications of the primitive functions `is_null`, `head`, `tail`, `pair`, and `list` do not create any frame.)

- A. 7
- B. 10
- C. 12
- D. 13
- E. 16
- F. 17
- G. 20
- H. None of the other options is the correct answer

(14) Of the environment frames that get created during the evaluation of Program C, how many extend the program environment *directly*? (Do not count the global environment frame. We assume that the applications of the primitive functions `is_null`, `head`, `tail`, `pair`, and `list` do not create any frame.)

- A. 3
- B. 4
- C. 7
- D. 8
- E. 11
- F. 15
- G. 19
- H. None of the other options is the correct answer

(15) Of the environment frames that get created during the evaluation of Program C, how many have bindings for the name `w`? (Do not count the global environment frame. We assume that the applications of the primitive functions `is_null`, `head`, `tail`, `pair`, and `list` do not create any frame.)

- A. 0
- B. 1
- C. 2
- D. 3
- E. 4
- F. 5
- G. 6
- H. More than 6

(16) How many function objects get created during the evaluation of Program C? (Do not count function objects of primitive and pre-declared functions such as `array_length`, `math_floor`, `pair`, `head`, `tail`, `length`, and `map`.)

- A.** 1
- B.** 2
- C.** 3
- D.** 4
- E.** 5
- F.** 6
- G.** 7
- H.** 8
- I.** More than 8

(17) Of the function objects that get created during the evaluation of Program C, how many have the program environment as their environment (i.e. their right circles point to the program frame)? (Do not count function objects of primitive and pre-declared functions such as `array_length`, `math_floor`, `pair`, `head`, `tail`, `length`, and `map`.)

- A.** 0
- B.** 1
- C.** 2
- D.** 3
- E.** 4
- F.** 5
- G.** 6
- H.** 7
- I.** More than 7

Section D

For all the questions in this section, consider the following Source program:

Program D:

```
function koo(fun, A) {  
    const len = array_length(A);  
    let m = 0;  
    while (m < len) {  
        A[m] = fun(A[m], A[m]);  
        m = m + 1;  
    }  
}  
  
function see(a, n) {  
    let acc = 0;  
    let i = 0;  
    while (i < n) {  
        let sum = acc + a;  
        acc = sum;  
        i = i + 1;  
    }  
    return acc;  
}  
  
const AA = [1, 2, 3, 4];  
koo(see, AA);  
AA;
```

(18) What is the result of evaluating Program D?

- A. 4
- B. 10
- C. 30
- D. [1, 2, 3, 4]
- E. [2, 4, 6, 8]
- F. [1, 3, 6, 10]
- G. [1, 4, 9, 16]
- H. None of the other options is the correct answer

(19) How many environment frames get created during the evaluation of Program D? (Do not count the global environment frame. We assume that the applications of the primitive function `array_length` do not create any frame.)

- A.** 11
- B.** 15
- C.** 17
- D.** 19
- E.** 21
- F.** 25
- G.** 29
- H.** None of the other options is the correct answer

(20) Of the environment frames that get created during the evaluation of Program D, how many extend the program environment *directly*? (Do not count the global environment frame. We assume that the applications of the primitive function `array_length` do not create any frame.)

- A.** 1
- B.** 2
- C.** 4
- D.** 5
- E.** 6
- F.** 9
- G.** 11
- H.** None of the other options is the correct answer

(21) Of the environment frames that get created during the evaluation of Program D, how many have bindings for the name `sum`? (Do not count the global environment frame. We assume that the applications of the primitive function `array_length` do not create any frame.)

- A.** 1
- B.** 2
- C.** 3
- D.** 4
- E.** 8
- F.** 10
- G.** 18
- H.** None of the other options is the correct answer

(22) How many function objects get created during the evaluation of Program D? (Do not count function objects of primitive and pre-declared functions such as `array_length`, `math_floor`, `pair`, `head`, `tail`, `length`, and `map`.)

- A.** 1
- B.** 2
- C.** 3
- D.** 4
- E.** 5
- F.** 6
- G.** 7
- H.** 8
- I.** More than 8

———— **END OF QUESTIONS** ————