

NATIONAL UNIVERSITY OF SINGAPORE

SCHOOL OF COMPUTING
MIDTERM ASSESSMENT FOR
Semester 1 AY2017/2018

CS2030 Programming Methodology II

October 2017

Time Allowed 80 Minutes

INSTRUCTIONS TO CANDIDATES

1. This assessment paper contains 9 questions and comprises 10 printed pages, including this page.
2. The last two pages of this assessment paper is the answer sheet.
3. Write all your answers in the answer sheet. Detach the answer sheet for submission at the end of the assessment.
4. The total marks for this assessment is 40. Answer **ALL** questions.
5. This is an **OPEN BOOK** assessment.
6. All questions in this assessment paper use Java 8 unless specified otherwise.
7. State any additional assumption that you make.

Part I**Multiple Choice Questions (12 points)**

- For each of the questions below, select the most appropriate answer and **write your answer in the corresponding answer box on the answer sheet..** Each question is worth 3 points.
- If multiple answers are equally appropriate, pick one and write the chosen answer in the answer box. Do NOT write more than one answers in the answer box.
- If none of the answers are appropriate, write x in the answer box.

1. (3 points) Consider the following definition of class A:

```
class A extends P implements I {  
}
```

Which of the following statements would result in a compilation error?

- (i) Object a = new A();
 - (ii) I a = new P();
 - (iii) I a = new A();
 - (iv) A a = new I();
- A. Only (i)
 - B. Only (iii)
 - C. Only (ii) and (iii)
 - D. Only (i) and (iv)
 - E. Only (ii) and (iv)

Write x in the answer box if none of the combinations is correct.

2. (3 points) Consider the program below:

```
class A {  
    static void f() throws Exception {  
        try {  
            throw new Exception();  
        } finally {  
            System.out.print("1");  
        }  
    }  
  
    static void g() throws Exception {  
        System.out.print("2");  
        f();  
        System.out.print("3");  
    }  
  
    public static void main(String[] args) {  
        try {  
            g();  
        } catch (Exception e) {  
            System.out.print("4");  
        }  
    }  
}
```

What is the output of the program?

- A. 24
- B. 214
- C. 23
- D. 213
- E. 2134

Write x in the answer box if none of the answers above is correct.

3. (3 points) Consider the program below:

```
class Main {  
  
    static class P {  
        public void f(int i, int j) {  
            System.out.print(i + j);  
        }  
    }  
  
    static class A extends P {  
        public void f(Integer i, Integer j) { // Note the type  
            System.out.print(i * j);  
        }  
    }  
  
    static class B extends P {  
        public void f(int j, int i) { // Note the order of i and j  
            System.out.print(i - j);  
        }  
    }  
  
    public static void main(String[] args) {  
        P objects[] = { new P(), new A(), new B() };  
        for (P p : objects) {  
            p.f(1, 2);  
        }  
    }  
}
```

What is the output of the program?

- A. 321
- B. 331
- C. 221
- D. 222
- E. 333

Write x in the answer box if none of the combinations is correct.

4. (3 points) Consider a method declared as `int foo(double x)`.

Which of the following statements will NOT result in a compilation error:

- (i) `int cs = foo(2030);`
- (ii) `double cs = foo(2.030);`
- (iii) `int cs = foo(new Double(2.030));`
- (iv) `Integer cs = foo(new Double(2.030));`

- A. (i) and (iv) only
- B. (ii) and (iv) only
- C. (ii) and (iii) only
- D. (i), (ii), and (iii) only
- E. (i), (ii), (iii), and (iv)

Write x in the answer box if none of the combinations is correct.

Part II

Short Questions (28 points)

Answer all questions in the space provided on the answer sheet. Be succinct and write neatly.

5. (3 points) **LSP.** Consider the following classes: `FormattedText` adds formatting information to the text. We call `toggleUnderline()` to add or remove underlines from the text. A `URL` is a `FormattedText` that is always underlined.

```
class FormattedText {
    public String text;
    public boolean isUnderlined;

    public void toggleUnderline() {
        isUnderlined = (!isUnderlined);
    }
}

class URL extends FormattedText {
    public URL() {
        isUnderlined = true;
    }
    public void toggleUnderline() {} // do nothing
}
```

Does it violate the Liskov Substitution Principle? Explain.

6. (6 points) **Type.** Explain how each of the following language features of Java ensures type safety. You can give an example.
- (a) (3 points) enum
 - (b) (3 points) generics

7. (8 points) **Wildcards.** For each of the statement below, indicate if it is a valid statement (no compilation error). Explain why in one sentence.

- (a) (2 points) `List<?> list = new ArrayList<String>();`
- (b) (2 points) `List<? super Integer> list = new List<Object>();`
- (c) (2 points) `List<? extends Object> list = new LinkedList<Object>();`
- (d) (2 points) `List<? super Integer> list = new LinkedList<>();`

8. (3 points) **Nested.** Consider the program below:

Note: The topic of nested class is not within the scope of AY 21/22 Sem 2 midterm.

```
class B {
    void f() {
        int x = 0;

        class A {
            int y = 0;
            A() {
                y = x + 1;
            }
        }

        A a = new A();
    }
}
```

Suppose that a variable `b` is an instance of class `B`, and a program calls `b.f()`. Sketch the content of the stack and heap immediately after the Line `A a = new A()` is executed. Label the values and variables / fields clearly. You can assume `b` is already on the heap and you can ignore all other content of the stack and the heap before `b.f()` is called.

9. (8 points) **Bus.** You are asked to write a program that prints out, given a bus stop, the list of bus services that serves the bus stop. Bus services and bus stops are identified by names. A name is a string that does not contain any space.

The query of which bus services serve a bus stop, however, is not readily available. You are given instead, a file that lists which bus stops a bus service serves. The file contains one bus service per line. The first name is the name of the bus service, the rest of the line contains the name of the bus stops. We assume that each bus stop appears only once per bus service (no repeats).

For example, the following shows the first few lines of a sample input file for NUS internal shuttle buses:

```
A1 PGP KR-MRT LT29 UHall OppUHC YIH CL LT13 AS7 COM2 BIZ2 OppH12 H7
A2 PGP H17 H12 OppHSS OppNUSS COM2 Ventus CCE OppYIH Museum UHC OppUHall S16 OppKR-MRT
B1 KR-Bus CCE OppYIH UTown YIH CL LT13 AS7 BIZ2
C KR-Bus KentVale Museum UHC OppUHall S17 LT29 UHall OppUHC RafflesHall EA
D1 OppHSS OppNUSS COM2 Ventus CCE OppYIH Museum UTown YIH CL LT13 AS7 COM2 BIZ2
:
```

If you are given an input "COM2", the program should print "A1 D1 A2", since COM2 is served by these three bus services. The order of the bus services printed is not important. Of course, your program should work for any other bus companies (e.g., SBS or SMRT buses).

- (a) (4 points) Which class or classes from Java Collections Framework would you use to store the information read from the input file so that you can answer the query "which bus services serve this bus stop?" *efficiently*? Explain why you choose them and how you would use them. You only need to choose from ArrayList, LinkedList, HashSet, HashMap, and PriorityQueue.
- (b) (4 points) Write two classes, in Java syntax, that you would use to encapsulate the information read from the input file and the operations necessary to answer the query "which bus services serve this bus stop?". In writing down the classes, you can skip:
- the implementation of the methods (just write { . . . })
 - private methods
 - methods and fields related to input and output

END OF PAPER

This page is intentionally left blank.

Matriculation Number:

--	--	--	--	--	--	--	--	--

1. 2. 3. 4.

5. LSP.

.....

.....

.....

.....

6. Type.

(a)

.....

.....

.....

.....

.....

.....

(b)

.....

.....

.....

.....

.....

.....

7. Wildcards.

(a)

.....

(b)

.....

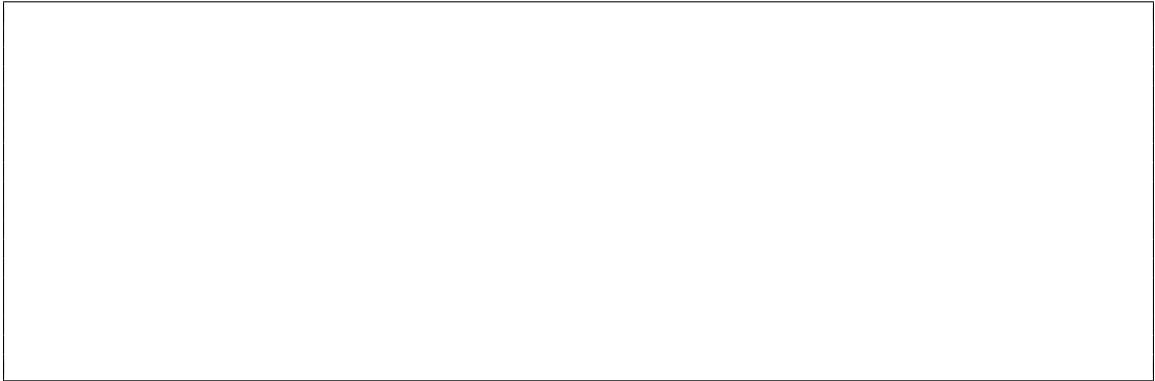
(c)

.....

(d)

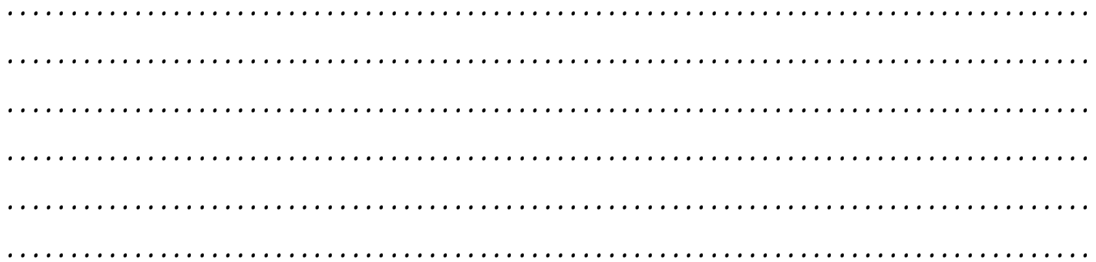
.....

8. Nested.



9. Bus.

(a)



(b)

