Student Name (Last name, First name): _____ Student ID: _____

THE HONG KONG POLYTECHNIC UNIVERSITY

DEPARTMENT OF COMPUTING

EXAMINATION

Course : Broad Discipline of COMP-61431 /
Broad Discipline of COMP-61431, BSc AB with Biotechnology-12451

Subject : COMP2021 Object-Oriented Programming

Group : 1011 / 1012, 1411

Session : 2022 / 2023 Semester I

Date : 10 December 2022 Time : 12:30 - 14:30

Time Allowed: 2 Hours Subject Lecturer: Dr PEI Yu Max /
Dr XU Linchuan

This question paper has 11 pages (cover included).
(Some pages may be intentionally omitted.)

Instructions to Candidates:

- This is a closed-book exam.
- Use a pen (not a pencil)! Answers written with a pencil will not be marked!!!
- Write your answers on this question paper.
- Write down your name and student numbers on each page on this question paper.
- Please write legibly!

Do not turn this page until you are told to do so!

Section B. Short-Answer Questions (36 points)

Question 6. (Information Hiding, 8 points) Given two packages p and q as well as five public classes A, B, C, D, and E, satisfying the following conditions:

p (A, B, C)
q. (D, E)

- q is a sub-package of p;
- classes A, B, and C are defined in package p;
- classes D and E are defined in package q; and
- both classes C and D inherit from class A.

Suppose class A has two instance methods m1 and m2 as listed in the table below, please fill **Yes** or **No** in each empty cell of the table to indicate whether the corresponding method is accessible in classes B, C, D, and E. (1 point x 8)

Methods in `p.A`	`p.B`	`p.C extends p.A`	`q.D extends p.A`	`q.E`
`protected void m1()`	✓	✓	✓	✗
`void m2()`	✓	✓	✓	✓

?

Question 7. (Method Binding, 8 points) Please 1) explain the differences between dynamic method binding and static method binding in terms of a) the time when each of them takes place and b) the kind of type information about the receiver that is exploited for resolving a call (4 points) and 2) answer which type of binding is used to resolve calls to static methods and which type is used to resolve calls to overridden instance methods (4 points).

啥时候学的?
when
time

dyna
继承
method.

Static
member?
不会

information

啥意思

记不清了就考这个...

Student Name (Last name, First name): _____ Student ID: _____

Question 8. (Polymorphism and Exception Handling, 10 points) Listing 1 defines two exception classes, namely `EA` and `EB`, and a utility class `U`.

Listing 1. Exception and Utility Classes.

```
public class EA extends Throwable { }
public class EB extends RuntimeException { }
public class U {
    public static void log(int id){ System.out.print(id + "."); }
    public static void h1() throws EA {  throw new EA();  }
    public static void h2() throws EB {  throw new EB();  }
}
```

1) Please answer a) whether `EA` is a checked or unchecked exception and b) whether `EB` is a checked or unchecked exception. (2 points x 2)

2) Listing 2 defines two classes `A` and `B` based on the exception and utility classes.

Listing 2. Classes A and B.

```
public class A {
    public A(){ U.log(1); }
    public A(int x){ U.log(2); }
    public void m1 () throws EA, EB { U.log(3); U.h1(); U.log(4); }
    public void m2() {
        try { U.log(5); m1(); U.log(6); }
        catch(Exception e){ U.log(7); }
        catch(Throwable e){ U.log(8); }
        finally{ U.log(9); }
    }
}
public class B extends A {
    public B(){ U.log(10); }
    public B(int x){ super(); U.log(11); }
    public void m1() throws EA, EB { U.log(12); U.h2(); U.log(13); }
    public void m2() {
        try { U.log(14); super.m2(); U.log(15); }
        catch(Exception e){ U.log(16); }
        catch(Throwable e){ U.log(17); }
        finally{ U.log(18); }
    }
}
```

For each of the test methods below, answer whether the test method compiles successfully. If yes, give the output produced by the test when it is executed; If no, *add the necessary code to the method body so that the test compiles successfully.* (2 points x 3)

```
@Test public void test1(){
    A a = new A(1);
    a.m2();
}
```

```java
@Test public void test2(){
    A a = new B(2);
    a.m2();
}
```

```java
@Test public void test3(){
    A a = new B(1);
    a.m1();
}
```

Question 9. (Concurrency, 10 points) (1) We can define a thread class in Java by extending class `java.lang.Thread` or by implementing interface `java.lang.Runnable`. Please explain why the second option is in general preferable to the first. (2 points)

(2) Given a running thread `t`, although we can terminate `t` by simply calling `t.stop()` or `t.destroy()`, a better way is to use interrupts. Please explain a) the problems with terminating threads using `t.stop()` or `t.destroy()` and b) how interrupts should be used to terminate threads. (4 points x 2)

过时了

(会弄坏?)

implement

仍不实现 run

会报错提醒,

但 Thread 不override

没事 (?存疑)

、interrupts,

(try catch)

、没明白问的啥

~~[illegible]这种~~

Student Name (Last name, First name): _____ Student ID: _____

Section C. Programming (44 points)

Question 10 (Generics, 17 points). A set is a collection of distinct objects. Every element of a set must be unique in that set (as determined by calls to method `equals` on set elements) and all set operations preserve this property. The order in which the elements are included in the set is irrelevant. Therefore, two sets {1, 2, 3} and {2, 3, 1} are considered equivalent.

Define a generic class `Set` with one formal type parameter `T`. Code snippet in Listing 4 specifies all the operations, including `has`, `add`, `addAll`, `remove`, `size`, and `toString`, that class `Set` must support. Note `Integer` is a sub-type of `Number`.

Listing 4. Operations that class Set must support.

```
Set<Number> s = new Set<>();       // create an empty set of Number objects
System.out.println(s.toString()); // print, e.g., "{}"
s.add(7);                          // s = {7}
System.out.println(s.size());      // print: 1
s.add(9);                          // s = {7,9}
s.add(null);                       // s = {7,9}
s.add(7);                          // s = {7,9}
System.out.println(s.has(9));      // print "true"
System.out.println(s.has(8));      // print "false"
s.add(8);                          // s = {7,9,8}
s.remove(7);                       // s = {9,8}
s.remove(4);                       // s = {9,8}
Set<Integer> s1 = new Set<>();     // s1 = {}
s1.add(6);                         // s1 = {6}
s1.add(8);                         // s1 = {6,8}
s.addAll(s1);                      // add all elements in s1 to s. s = {9,8,6}
System.out.println(s.toString()); // print, e.g., "{9,8,6}"
```

Please design and implement class `Set` to meet the requirements.

模拟

Student Name (Last name, First name): _____ Student ID: _____

Question 11 (Object-based programming, 17 points). A network consists of a set of nodes and a set of edges. In this question, you may assume all the input networks satisfy the following conditions: 1) Each edge connects two different nodes; 2) Two nodes can only be connected by a single edge; 3) Each node is connected to at least one other node. Listing 3 gives snippets of two classes Network and Node, and you need to complete the following tasks:

1. To add a constructor of Network that constructs from a list of edges a two-dimensional array representation of the network, stored in a field named adjacencyMatrix. The list of edges is given as List<String>, where each element has the format "nodeID$_1$,nodeID$_2$", and each nodeID is an integer from {0, 1, 2, ..., N-1} (N is the number of nodes). Two nodes x and y are connected if and only if both adjacencyMatrix [x][y] and adjacencyMatrix [y][x] are equal to 1 ($0 \le x, y \le N-1$). Figure 1 gives an example list of edges and the adjacencyMatrix constructed from the edges.

```
List of edges:
  ["0,1", "1,2"]
adjacencyMatrix:
  {{0,  1,  0},
   {1,  0,  1},
   {0,  1,  0}}
```

Figure 1. An example list of edges and the resulting adjacencyMatrix.

2. To define a method setDegreeCentrality of Node to calculate and set the degree of a node. The degree of a node is the number of nodes it is connected to.
3. To override the hashCode method of Node so that two equivalent Node objects always return the same hash code.
4. To override the equals method of Node so that two Node objects are considered equal if and only if they belong to the same network and have the same node ID.
5. To implement method compareTo of Node so that, for each pair of nodes n1 and n2 (n1 != null && n2 != null):

   n1.compareTo(n2) > 0 if n1's degree centrality value is larger than n2's;

   n1.compareTo(n2) == 0 if n1's degree centrality value is equal to n2's;

   n1.compareTo(n2) < 0 if n1's degree centrality value is smaller than n2's;

Note: 1) You are not allowed to add new methods. 2) You may assume that relevant libraries have been properly imported. Besides, you may find the following methods from the standard Java library useful:

Class String:

int indexOf(char c): Returns the index within this string of the first occurrence of the specified character.

int length(): Returns the length of this string.

String substring(int beginIndex, int endIndex): Returns a new string that is a substring of this string.

Class Integer:

static int parseInt(String s): Parses the string argument as a signed decimal integer.

Listing 3. Classes Network and Node.

```
public class Network {
    private int[][] adjacencyMatrix;
    public int[][] getAdjacencyMatrix(){ return adjacencyMatrix; }

    public Network(List<String> edgeList){
        // complete this method (6 points)
        // All strings contained in 'edgeList' are in the format "nodeid,nodeid"
```

模拟2(懒惰的写

Student Name (Last name, First name): _____ Student ID: _____

Listing 9. Class PrimitiveItemCounter.

```
public class PrimitiveItemCounter implements GraphItemVisitor{

    // add necessary code to the class (2 points)

    cnt = 0

    public void visit(Square s){
        // complete this method (2 points)
        cnt ++

    }

    public void visit(Circle c){
        // complete this method (2 points)
        cnt ++

    }

    public void visit(Graph graph){
        // complete this method (2 points)
        for (gi : graph.getItems())
            if (gi instanceof Square or Circle)
                visit(gi)

    }

    public int getPrimitiveItemNbr(){
        // complete this method (2 points)
        return cnt;

    }
}
```

没明白为啥不是每个函数一行

** END **