

NATIONAL UNIVERSITY OF SINGAPORE

SCHOOL OF COMPUTING MIDTERM TEST FOR CS1020 AY2011/12 Semester 2

CS1020 – Data Structures and Algorithms I

2 March 2012

Time allowed: 1 hour 30 minutes

Matriculation
number:

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

INSTRUCTIONS TO CANDIDATES

1. This test paper consists of **FIFTEEN (15)** questions and comprises **THIRTEEN (13)** printed pages.
2. This is a **CLOSE BOOK** test. You are allowed to bring in ONE (1) piece of **handwritten** A4 reference sheet (no photocopies).
3. Fill in your Matriculation Number above clearly with a pen. Note that your matriculation number contains a letter at the back, for example: U084321X or A0091234E.
4. Answer all questions.
5. For MCQs (Q1 to Q12), use the OCR form provided. Shade and write down your matriculation number on the OCR form. You must use 2B pencil to shade/write on the OCR form, or the grading machine might not be able to register your shading.
6. For short questions (Q13 to Q15), fill in your answers in the space provided. You may use pencil or pen to write your answers.
7. You must submit both the OCR form and this document. It is your responsibility to ensure that you have submitted both to the invigilator at the end of the test.

EXAMINER'S USE ONLY			
Section / Question	Possible	Marks	Check
A. MCQ 1-12	48		
B. Q 13	26		
B. Q 14	16		
B. Q 15	10		
Total	100		

SECTION A (12 Multiple Choice Questions: 48 Marks)

Each question has one correct answer. Write your answer in the space provided on the OCR form. 4 marks are awarded for each correct answer; no penalty for wrong answer.

1. What is the output of the following line of code?

```
System.out.println("pi to two digits is " + (314/100));
```

- a. pi to two digits is 3
- b. pi to two digits is 3.14
- c. pi to two digits is 314/100
- d. pi to two digits is 314
- e. None of the Above

2. What is printed as a result of the following code?

```
class TestSwap {  
  
    public static void swap(int[] a1, int[] a2) {  
        int[] temp = a1;  
        a1 = a2;  
        a2 = temp;  
    }  
  
    public static void main(String[] args) {  
        int[] A1 = {2, 3};  
        int[] A2 = {4, 5};  
        swap(A1, A2);  
        System.out.println "[" + A1[0] + ", " + A1[1] + " ]";  
    }  
}
```

- a. [2, 3]
- b. [4, 5]
- c. [3, 2]
- d. [2, 5]
- e. [2, 4]

3. Which of the following things are TRUE about a static attribute?

- i. each object gets its own copy
- ii. is only useable inside a static method
- iii. must be initialized inside a constructor

- a. (i)
- b. (ii)
- c. (iii)
- d. (ii) and (iii)
- e. None of (i) (ii) (iii)

4. What does the following code print?

```
class A {
    private int a = 0;
    public A() {
        a=5;
    }
    public A(int a) {
        a=a;
    }
    public int get() {
        return a;
    }
}

class Test {
    public static void main(String[] args) {
        A a1 = new A();
        A a2 = new A(5);
        A a3 = new A(10);
        System.out.println(a1.get() + " " +
                           a2.get() + " " + a3.get());
    }
}
```

- a. 5 0 0
- b. 5 5 10
- c. 0 5 10
- d. 0 0 0
- e. 5 5 5

5. What does the following code print?

```
class A {
    public void print() {
        System.out.print("A");
    }
}

class B extends A {
    public void print() {
        System.out.print("B");
    }
}

class Test {
    public static void main(String[] args) {
        A a1 = new A();
        A a2 = new B();
        B b1 = new B();
        A a3 = (A) b1;
        B b2 = (B) a3;

        a1.print();
        a2.print();
        a3.print();
        b1.print();
        b2.print();
    }
}
```

- a. ABBBB
- b. AAABB
- c. ABABA
- d. ABABB
- e. AABBB

6. Given two classes as follows:

```
class A {
    private int A_i;

    //other code not shown
    public void print() {
        System.out.println(A_i);
    }
}

class B extends A {
    private int B_i;

    //other code not shown
    public void print() {
        //Implementation
    }
}
```

Which of the following implementation(s) for class B's print() method correctly print out the value of object attribute **A_i** followed by **B_i**?

- i. `System.out.println(A_i);`
`System.out.println(B_i);`
 - ii. `super.print();`
`System.out.println(B_i);`
 - iii. `super.print();`
`this.print();`
- a. (i) only.
 - b. (ii) only.
 - c. (i) and (iii) only.
 - d. (ii) and (iii) only.
 - e. (i), (ii) and (iii).

7. Given the following code:

```
interface betterThan {
    boolean betterThan(betterThan bt);
}

class Shape implements betterThan {
    private int _num_sides;
    public Shape(int sides) {
        _num_sides = sides;
    }
    public boolean betterThan(betterThan bt) {
        Shape s = (Shape) bt;
        return (_num_sides < s._num_sides);
    }
}
```

What is NOT a possible result of calling the **betterThan** function?

- i. returns true when the argument has more sides than “this” Shape
 - ii. returns false when the argument has more sides than “this” Shape
 - iii. raises a runtime error (exception)
- a. (i)
 - b. (ii)
 - c. (iii)
 - d. (i) and (iii)
 - e. (ii) and (iii)

8.

```
public void Test() {
    int i = 0;
    try { // both f and g can raise exceptions
        f(); i++; g(); i++;}
    catch(Exception e) {i--;}
    finally {i--;}
    System.out.print(i);
}
```

What output value is NOT possible from running Test?

- a. 1
- b. 0
- c. -1
- d. -2
- e. All of the above are possible

```

9. class Tuple<T> {
    private T[] my_data;

    public Tuple(T[] data) {
        my_data = data;
    }

    public T project(int i) throws OutOfBoundsException {
        if (i < 0 || i >= my_data.length)
            throw (new OutOfBoundsException("bad index"));
        return my_data[i];
    }

    public String toString() {
        String s = "<";
        for (T myT : my_data) s += myT + ",";
        return s + ">";
    }
}

```

What is the result of running the following statements?

```

x = new Tuple<int>({3,4,5});
y = new Tuple<Tuple<int>>({x,x,x,x});
System.out.print(y.project(1).project(1) + "#"+ y.project(1));

```

- 3#<3, 4, 5>
- 3#<3, 4, 5, >
- 4#<3, 4, 5>
- 4#<3, 4, 5, >
- 3#3

```

10. interface toInt {
    int toInt();
}

```

```

class myData implements toInt {...}

```

What must be true about **myData**?

- Implements a private method toInt() returning an int.
- Implements a public method toInt() returning an int.
- Implements a public method toInt(int x) returning an int.
- Implements a public method toInt() returning an Integer.
- Can be empty (no attributes or methods)

11. Which of the following is NOT true regarding using an ADT to design a program component?

- i. The ADT indicates which implementation is preferred.
- ii. The ADT specifies the running time for its methods.
- iii. The ADT specifies which methods the user of the component should expect.

- a. (i) only
- b. (ii) only
- c. (iii) only
- d. Two of (i), (ii), (iii)
- e. All three of (i), (ii), (iii)

12. Which of the following is NOT true about object oriented programming?

- i. An object is a package with both data and the methods that operate on that data.
- ii. Inheritance is a way to extend an existing type of objects with new attributes and/or methods.
- iii. All attributes must be private.
- iv. Static attributes and methods “belong” to the class and not to any particular instantiation (object) of the class.

- a. (i) only
- b. (ii) only
- c. (iii) only
- d. (iv) only
- e. Two of (i), (ii), (iii), (iv)

SECTION B (3 Short Questions: 52 Marks)**Question 13 (18 + 8 = 26 total marks)**

Recall the concept of a linked list from class. One simple implementation (which does not follow the encapsulation principle) is given here:

```
class Node {
    public int data;
    public Node next;
    public Node(int data, Node next)
        { this.data = data; this.next = next; }
}

class List {
    public Node head;
    public List(Node head) {this.head = head;}
    public List pivot() { ... }
}
```

The **pivot()** function takes a list A and splits it into two lists B and C such that everything in B is less than the first element in A, and such that everything in C is greater than or equal to the first element in A, and moreover the first element of A is also the first element of C. Except for these rules, the order of B and C do not matter. The function then returns B and sets “this.head” to C.

For example, if the input list A = {4,6,2,6,4,1,5}, then one choice for B (returned from the function) = {2,1} and C (equal to this.head afterwards) = {4,6,6,4,5}. Since we do not care about order (except for the first element of C), it would be fine if B = {1,2} and C = {4,4,5,6,6}.

Please fill in the blanks in the code on the next page to implement **pivot()**. You can earn a maximum of 18 marks if your solution constructs additional nodes (i.e., **new Node(...)**). For full credit (26 marks), do not construct any additional nodes.

```
public List pivot() {
    if (head == null) return new List(null);

    int p = head.data; // p is the pivot value
    Node B = null; // B points to the new list.

    return (new List(B));
}
```

Question 14 (16 marks)

A car dealer sells three kinds of cars: Toyota Corolla, Honda Civic, and Honda NSX. (Note that Toyota and Honda are car brands (car companies); Corolla, Civic, and NSX are car models produced by the car companies.) All three kinds of cars share certain basic functionality (e.g., all three can be driven). On the other hand, there are some features unique to Toyotas, and other features unique to Hondas (e.g., the logo of the car company). Finally, each model of car is different (so a Civic has some features that an NSX does not and vice versa).

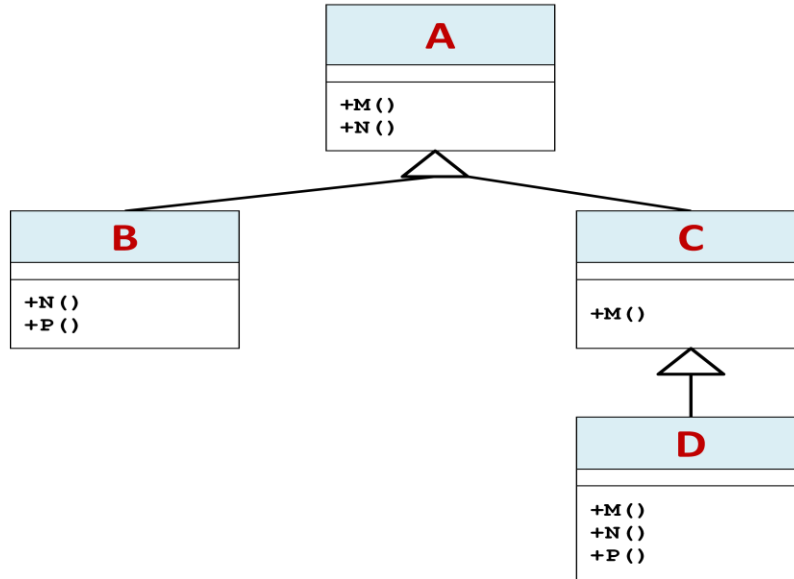
(Part A, 8 marks) Write some code that defines a series of classes that incorporate these relationships. We are not interested in the class bodies (use { ... } instead), but rather your ability to express logical relationships using object oriented principles.

(Part B, 8 marks) Assume that each car has a maximum speed “public double MAX_SPEED”. Based on your design in part (a), implement a **static method** that:

- Takes in **an array of cars** (regardless of brand or model).
- Returns the arithmetic mean (average), of type double, of the speeds of the cars in the array.

Question 15 (10 marks)

Given the following class hierarchy in UML notation:



For simplicity, you can assume that all methods in the various classes simply prints out message of the form `<class_name>.<method name>`, e.g. the method `M()` in class `A` prints out `"A.M"`. Note also that if a class overrides an inherited method, the method's name will appear in the class icon, e.g. class `B` overrides the method `N()`. Otherwise, the inherited method remains **unchanged** in the subclass.

For each of the code fragments in the next page, indicate whether:

- The code will cause compilation error. Briefly explain the reason. **OR**
- The code compiles and executes. Give the execution result.

Here are two examples for your reference:

Code Fragment (Example)	Compilation Error	Execution Result
<code>A a = new A();</code> <code>a.N();</code>		A.N
<code>C c = new C();</code> <code>c.P();</code>	method P() not defined in class C	

Note that any code fragment with both choices filled will be taken as incorrect. Each code fragment should be considered on its own, independent from other code fragments.

Code Fragment (Example)	Compilation Error	Execution Result
<code>A a = new B (); a.N ();</code>		
<code>B b = new A (); b.N ();</code>		
<code>C c = new D (); c.P ();</code>		
<code>A a = new C (); a.N ();</code>		

~~~ END OF PAPER ~~~