NATIONAL UNIVERSITY OF SINGAPORE

SCHOOL OF COMPUTING

MIDTERM TEST FOR
Semester 2, AY2016/17

CS1020 – Data Structures and Algorithms 1

March 2017 Time allowed: 1.5 hours

MATRICULATION NO. : | | | | | | | | | |

(Write your Matriculation Number legibly with a pen.)

INSTRUCTIONS TO CANDIDATES

1. Write your matriculation number in the space provided above.
2. This paper consists of 8 MCQ questions, 3 short questions and 1 programming question. It comprises ten (10) printed pages including this front page.
3. Answer all questions. Use the provided OCR form for the 8 MCQ questions, and write your answer for the short questions and programs directly in the space given after each question.
4. Use 2B or darker pencil to shade the OCR form. You are allowed to use PENCIL to write your answers in this question papers.
5. You must submit both the OCR form and this paper. It is your responsibility to ensure that you have submitted both to the invigilator at the end of the test.
6. Marks allocated to each question are indicated. Total marks for the paper is 30.

Question	***Max***	***Marks***
Q1 – Q10	40	From OCR
Q11 – Q13	18	
Q14	24	
Q15	18	
Total	**100**	

Q1 (4 Marks) Which of the following statements is true about objects (in Java language)

a) Object can **ONLY** be created using the **new** keyword together with the constructor.
b) Once we have created an object of class A for a variable X, this variable X cannot be used to be assigned another object of class A.
c) Once we have created an object of class A for a variable X, this variable X cannot be used to be assigned an object of another class B.
d) Each object can use only methods defined within the class of this object.
e) None of the above.

Q2 (4 Marks) Which of the following statement(s) is/are true about Java:

(i) Subclass always has equal or more attributes than its superclass.
(ii) Subclass always has equal or more number of constructors than its superclass.
(iii) Once a method is overridden by a subclass, the method (of the same name) in the superclass cannot be used by the subclass.

a) i only
b) ii only
c) iii only
d) i and ii only
e) i and iii only

Q3 (4 Marks) Which of the following statement(s) is/are true about Java:

(i) Subclass of an abstract class must not be yet another abstract class.
(ii) A class can extend from only one abstract class, but can implement many interfaces.
(iii) A class can implement only one interface, but can extend many abstract classes.

a) i only
b) ii only
c) iii only
d) i and ii only
e) i and iii only

Q4 (4 Marks) Which of the following is true for object-oriented problem solving:

a) Within a class, we seek to have less coherence; among two or more classes, we seek to have less coupling.
b) Within a class, we seek to have less coherence; among two or more classes, we seek to have more coupling.
c) Within a class, we seek to have more coherence; among two or more classes, we seek to have less coupling.
d) Within a class, we seek to have more coherence; among two or more classes, we seek to have more coupling.
e) Coherence and coupling are not important in OO problem solving.

Q5 (4 Marks) Which of the following statement(s) is/are true about Java:

(i) We can choose to use the same name for two different methods from two different classes.
(ii) Within a class, we can use the same name for two methods only if they have different number of parameters.
(iii) Among methods of the same name, it is decided during compilation which method is actually being called.

a) i only
b) ii only
c) iii only
d) i and ii only
e) i and iii only

Q6 (4 Marks) What can be the possible output of the following statement where b is a double variable:

```
System.out.println( Math.floor(b) – Math.ceil(b + 0.5));
```

a) 1.0
b) 1.0 or 2.0
c) -1.0
d) -1.0 or -2.0
e) None of the above

Q7 (4 Marks) What is the output of the following code?

```
class TestQ7 {
     public static void main (String[] args) {
          int a[] = {7, 3, 2, 5, 4, 1, 6};
          int temp;

          for (int i=1; i < a.length; i++)
               if (a[i] < a[i-1]) {
                    temp = a[i];
                    a[i] = a[i-1];
                    a[i-1] = temp;
               }

          int sum=0;
          for (int i=1; i < a.length; i++)
               sum += a[i]*i;

          System.out.println("sum is " + sum);           )
}
```

a) 84
b) 88
c) 100
d) 140
e) None of the above

Q8 (4 Marks) What is the output of the following code?

```
class TestQ8 {

     public static void main (String[] args) {
          int[][] b = { {4,5,2}, {1,3}, {7,1,5,6} };
          int sum = 0;

          for (int i=0; i < b.length; i++)
               for (int j=0; j < b[i].length; j++)
                    sum += b[i][j] * i;

          System.out.println(sum);
     }
}
```

a) 11
b) 34
c) 42
d) 76
e) Compilation error and thus no output

Q10. (4 Marks) 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

Q11 (6 Marks) What are the outputs from the following codes:

```
public class TestString {
     public static void main (String[] args) {
          System.out.println("-" + 2 + 3);
          System.out.println(2 + "-" + 3);
          System.out.println(2 + 3 + "-");
     }
}
```

Answer: _____

_____

_____

Q12 (6 Marks) What are the outputs from the following codes:

```
public class Reconcile {

     private static int code = 8;
     private int a, b, c;

     public Reconcile (int a, int b, int c) {
          this.a = a;
          this.b = b = a;
          this.c = code;
          code++;
     }

     public Reconcile (int a, int b) {
          this(a, b, b);
     }

     public String toString() {
          return a + " " + b + " " + c;
     }

     public static void main (String[] args) {
          int a = 1;
          int b = 2;
          int c = 3;
          Reconcile d = new Reconcile (a, c, b);

          System.out.println("d = " + d);

          Reconcile e = new Reconcile (a, c);
          System.out.println("e = " + e);

     }

}
```

Answer:

_____

_____

Q13 (6 Marks) What are the outputs from the following codes:

```
class A {
     int f (int n) {
          return n+2;
     }

     int h (int n) {
          return n + f(n);
     }
}

class B extends A {
     int h (int n, A a) {
          return f( a.h(n));
     }
}

class C extends B {
     int g (int n) {
          return h(n – 6, this);
     }
}

public class D {

     public static void main (String[] args) {
          C c = new C();
          System.out.println ("c.h(2) = " + c.h(2) );
          System.out.println ("c.g(2) = " + c.g(2) );
     }

}
```

Answer:

_____

_____

Q14 (24 Marks) You are going to create a new class from our understanding of the Vector class in the java.util.* , and the (non-generic and modified) Pair class in our lecture. The following is reproduced from our lecture for your reference.

SYNTAX

```
//Declaration of a Vector reference
Vector<E> myVector;
//Initialize an empty Vector object
myVector = new Vector<E>;
```

Commonly Used Method Summary	
boolean	***isEmpty()*** Tests if this vector has no components.
int	***size()*** Returns the number of components in this vector.
boolean	***add*(E o)** Appends the specified element to the end of this Vector.
void	***add*(int index,E element)** Inserts the specified element at the specified position in this Vector.
E	***remove*(int index)** Removes the element at the specified position in this Vector.
boolean	***remove*(Object o)** Removes the first occurrence of the specified element in this Vector. If the Vector does not contain the element, it is unchanged.
E	***get*(int index)** Returns the element at the specified position in this Vector.
int	***indexOf*(Object elem)** Searches for the first occurrence of the given argument, testing for equality using the equals method**.**
boolean	***contains*(Object elem)** Tests if the specified object is a component in this vector.

```
class Pair {
     private String   _first;
     private Integer  _second;

     public Pair (String a, Integer b) {
          _first = a;
          _second = b;
     }
     public String  getFirst()  { return _first; };
     public Integer getSecond() { return _second; };
     public void incSecond() { _second ++; };
     public void decSecond() { _second --; };
}
```

Your task is to create a class called `VectorStrCnt`. A part of this class is completed for you in the following. Basically, this class is a vector that has Pair as an element in the vector. Each element is a pair that keeps a string and the corresponding times the string has been added to the vector.

```
import java.util.Vector;

class VectorStrCnt {
     private Vector<Pair> _strCnt;

     public VectorStrCnt() {
          _strCnt = new Vector<Pair> ();
     };
     public Boolean isEmpty() {
          return _strCnt.isEmpty();
     }
     public int size() {
          return _strCnt.size();
     }
     public Pair get(int i) {
          // to be completed by you (1 mark)
     };
     public boolean add(String s) {
          // to be completed by you (3 marks)
     };
     public boolean remove(String s) {
          // to be completed by you (2 marks)
     };
}
```

You are to complete the following 3 methods:

`public Pair get(int i)`
which simply returns the i^{th} element (pair) in the vector.

`public boolean add(String s)`
which first check whether the **`String s`** has already appeared in some pair in the vector. If yes, it increments the **`count`** for this found pair. If no, it adds **`String s`** to the end of the vector with **`count = 1`**.

`public boolean remove(String s)`
which first check whether the **`String s`** is in some pair in the vector. If no, return false. If yes, it decrements the **`count`** for this **`String s`**. If the **`count`** becomes zero, this pair is removed from the vector too.

Answer:

```
public Pair get(int i) {                          // 6 mark

……

……

……
}
```

```
public boolean add(String s) {                    //9 marks
```

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

```
}
```

```
public boolean remove(String s) {            //9 marks
```

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

```
}
```

Q15. (18 marks) We can use two BasicLinkedLists (BLL) to sort a given list of integers. Complete the following method which takes in a non-empty array of integers and return a BasicLinkedList sorted in ascending order. Some of the variables are given. You may recall that there are two important methods in BLL.

public void addFirst(E item);
public E removeFirst() throws NoSuchElementException;

```
public static BasicLinkedList <Integer> sort(int [] intList) {
  BasicLinkedList <Integer> list1 = new BasicLinkedList<Integer>();
  BasicLinkedList <Integer> List2 = new BasicLinkedList<Integer>();









}
```

=== END OF PAPER ===