

MULTIMEDIA



UNIVERSITY

STUDENT ID NO

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

SEAT NO: \_\_\_\_\_

SECTION: TC01

# MULTIMEDIA UNIVERSITY

## FINAL EXAMINATION

TRIMESTER 3, 2017/2018

### TCP1201 – OBJECT-ORIENTED PROGRAMMING AND DATA STRUCTURES

( All sections / Groups )

1 June 2018  
3:00 p.m. – 5:00 p.m.  
( 2 Hours )

Question	Mark
1	
2	
3	
4	
Total	

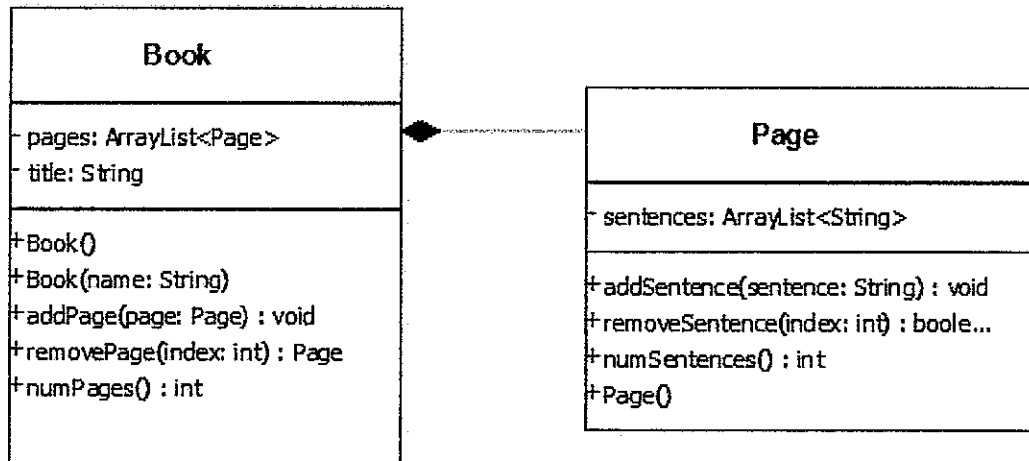
---

#### INSTRUCTIONS TO STUDENTS

1. This Question paper consists of 12 pages with 4 Questions only.
2. Attempt all **FOUR** questions. All questions carry equal marks and the distribution of the marks for each question is given.
3. Please write all your answers in **this Question Paper**.

**Question 1:****[ 25 marks]**

- a) Given the following UML class diagram.



- 1) Write the Java code for Page. You must write the body of the code for each method and initialize the sentences variable correctly. **[6 marks]**

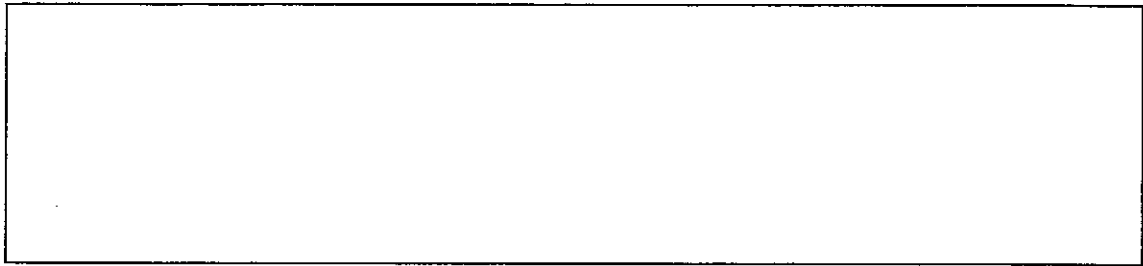
**Continued...**

- 2) Write the Java code for `Book`. If a new book is created without a title, the no-args constructor should give it the title "Unnamed Book" by calling the other constructor. The other constructor should initialize the pages variable correctly. [ 7 marks]

- 3) Write a Test class named *CheckBook*. In the *main()* method, create one sentences *s1* with the value "any string". Then create a Page *p1*. Add the sentence *s1* to page *p1*. Create a book *b* with the title "The one" and add page *p1* to it. Write a *println* statement to print the total number of pages in the book *b*. [6 marks]

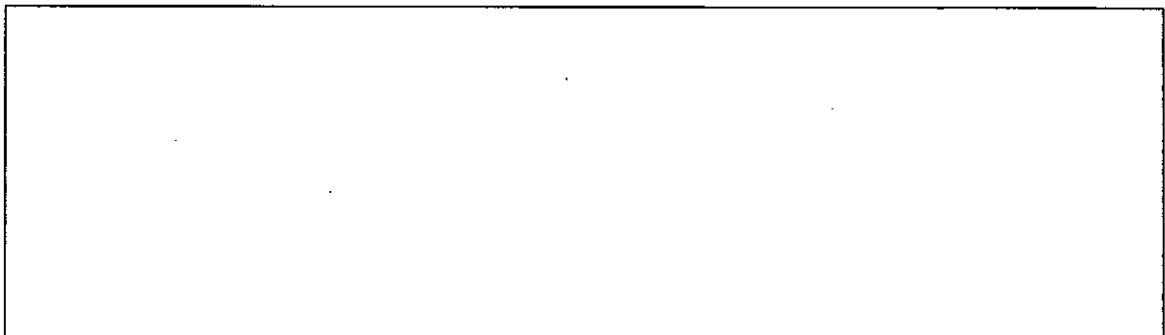
Continued...

- 4) Describe the type of relationship shown to exist between the *Page* and *Book* classes. [2 marks]



- b) What will be printed out after executing the following java program? [4 marks]

```
public class Strings1 {  
    public static void main(String[] args) {  
  
        String s1 = "Hello";  
        String s2 = "Hello";  
        String s3 = new String("Hello");  
        if (s1 == s2)  
            System.out.println("s1==s2");  
        else  
            System.out.println("s1!=s2");  
  
        if (s1 == s3)  
            System.out.println("s1==s3");  
        else  
            System.out.println("s1!=s2");  
    }  
}
```



Continued...

**Question 2:****[ 25 marks]**

Given the following Java Point class and Exam class:

```
public class Point implements Comparable<Point>{
    private double x;
    private double y;
    private double disOrigin;

    public Point (){
        this.x = 0;
        this.y = 0;
        computeDistance();
    }

    public Point (int x,int y){
        this.x=x;
        this.y=y;
        computeDistance();
    }

    private void computeDistance(){
        disOrigin = Math.sqrt(x*x+y*y);
    }
}
```

```
import java.util.Arrays;

public class Exam {
    public static void main(String[] args){
        public static void main(String[] args) {
            Point[] points = new Point[4];
            points[0] = new Point(1,3);
            points[1] = new Point(3,2);
            points[2] = new Point(-1,4);
            points[3] = new Point(-3,-2);

            Arrays.sort (points);
            for (Point point : points) {
                System.out.println(point);
            }
        }
    }
}
```

**Continued...**

- a. The main method in the Exam class will cause a compilation error. You must implement a missing method in the Point class to enable the sorting algorithm to work. What is the name of the method that you must implement and write a proper implementation for that method. **[6 marks]**

- b. In order for the line `[System.out.println(point);]` inside the `main()` method of the Exam class to properly print the sample output as shown below, write an appropriate implementation to override the `[toString()]` method for the Point class. **[3 marks]**

**Sample output**

(1.00,3.00-3.16) (3.00,2.00-3.61) (-3.00,-2.00-3.61) (-1.00,4.00-4.12)

**Continued...**

- c) Implement a proper **copy constructor** for the Point class. Then, write a java statement to invoke that **constructor** inside the main method and print out the point. [7 marks]

- d) Given the following class declarations and the Examq2 test class. Show the output produced after executing the program. If any of the statements in the main method causes a compilation error, identify that statement and correct it. [9 marks]

<pre>package examq2; public class Examq2 { static int a = 555; public static void main(String[] args) {     A objA = new A();     B objB1 = new B();     A objB2 = new B();     C objC1 = new C();     B objC2 = new C();     A objC3 = new C();     objA.display();     objB1.display();     objB2.display();     objC1.display();     objC2.display();     objC3.display(); } }</pre>	<pre>public class A {     int a = 100;     public void display() {         System.out.println("a in A="+a);     } }  public class C extends B {     private int a = 543;     public void display() {         System.out.printf("a in C="+a);     } }  public class B extends A{     private int a = 123;     public void display(){         System.out.printf("a in B="+a);     } }</pre>
---	---

Output:

Continued...

**Question 3:****[ 25 marks]**

- a. What is the output of the following code? And how many elements are left in the stack? **[5 marks]**

```

int [ ] values = {1,2,4,5,3,7,9,11};
Stack<Integer> s = new Stack<Integer>();
for (int i=0; i<values.length; i++)
    s.push(values[i]);
int n = 0;
for (int i=0; i<4; i++){
    n+=s.pop();
}
System.out.println(n);
for (int i = 0; i < 2; i++){
    n -= s.pop( );
}
System.out.println(n);
}

```

- b. Suppose that Q is an initially empty array-based queue of size 5. Show the values of the data members front and back after each statement has been executed. Indicate any errors that might occur.

**[6 marks]**

Statement	front	back
Queue<Character> Q(5);		
Q.enqueue ('A');		
Q.enqueue ('B');		
Q.enqueue ('C');		
char c = Q.dequeue();		
Q.enqueue ('A');		

**Continued...**



c. Given the following StackInterface:

[7 marks]

```
public interface StackInterface {  
    public void push(Object obj);  
    public Object pop();  
    public Object peek();  
}
```

And given the ArrayStack that implements the StackInterface:

```
1  import java.util.Vector;  
2  public ArrayStack {  
3      private Vector<Object> stackVector=new Vector<>();  
4      private final int INVALID_STACK_POINTER=-1;  
5      private int stackPointer==INVALID_STACK_POINTER;  
6      @Override  
7      public void push(Object obj){  
8          stackVector.add(++stackPointer, obj);  
9      }  
10     @Override  
11     public void pop() {  
12         Object obj = null;  
13         if (stackPointer != INVALID_STACK_POINTER) {  
14             obj = stackVector.elementAt(stackPointer);  
15             stackVector.removeElementAt(--stackPointer);  
16         }  
17     }  
18     @Override  
19     public Object peek() {  
20         Object obj = null;  
21         if (stackPointer != INVALID_STACK_POINTER) {  
22             obj = stackVector.elementAt(stackPointer);  
23         }  
24         return null;  
25     }  
26 }
```

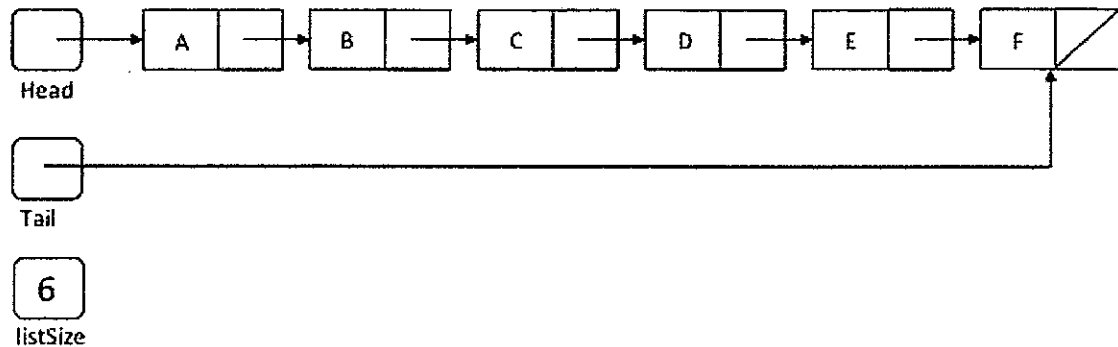
There are 7 syntax and logical errors in the class above, identify the line number and correct these errors.

--	--

Continued...

d. Given the following list:

[7 marks]



And the following method:

```

public void Shuffle( ){
    Node[] nodes = new Node[listSize];
    Node node = head;
    int i = 0;
    while (node != null) {
        nodes[i++] = node;
        node = node.next;
    }
    for ( i = 0 ; i < listSize ; i += 2 ) {
        nodes[i+1].next = nodes[i];
    }
    for ( i = (listSize - 1) ; i > 1 ; i -= 2 ) {
        nodes[i-3].next = nodes[i];
    }
    head = nodes[1];
    tail = nodes[listSize - 2];
    nodes[listSize-2].next = null;
}
  
```

Draw the list after **Shuffle** has finished executing.

Continued...

**Question 4:****[ 25 marks]**

- a. For each of the following scenarios **identify** the “best” **data structure** from the following list or a combination of **data structures**: an **unsorted array**, **linked list (LL)**, **Doubly LL**, **circular LL**, **stack**, **queue**. In each case, **justify** your answer **briefly**. **[3 marks]**

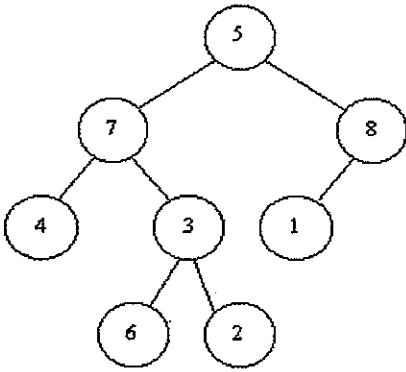
1. Suppose that a grocery store decided that customers who come first will be served first

2. A program needs to remember operations it performed in opposite order

3. The size of a file is unknown. The entries need to be entered as they come in. Entries must be deleted when they are no longer needed. It is important that structure has flexible memory management.

- b. Consider the following tree:

**[5 marks]**

	<p>1. Write out the order in which elements are visited if a PostOrder traversal is performed on the tree.</p> <p>2. Write out the order in which elements are visited if an InOrder traversal is performed on the tree.</p>
---	--

Continued...

- c. Given the following Java code that declares 2 types of sets and inserts the contents of an array (a) to them. Show the output generated. [4 marks]

```
int a[]={1,3,5,5,2,6,4};
HashSet<Integer> h = new HashSet();
LinkedHashSet<Integer> lh= new LinkedHashSet<Integer>();
TreeSet<Integer> th = new TreeSet(h);

for (int i=0;i<a.length;i++) lh.add(a[i]);
for (int i=0;i<a.length;i++) th.add(a[i]);

System.out.println(lh);//prints set contents between []
System.out.println(th);//prints set contents between []
```

```
[
[
```

- d. If we want to create a **binary tree** with nodes that contain integer values, we can represent the nodes using instances of the following Java class. [5 marks]

```
public class BTNode {
    public int value;
    public BTNode left;
    public BTNode right;
}
```

Write a recursive implementation of the following (**sum**) method so that it returns the sum of the values contained in all of the nodes of the binary search tree with **root n**.

```
public int sum(BTNode n){

}

}
```

Continued...

- e. One common way to implement a list is a single-linked list containing a collection of nodes that refers to the data objects in the list. We can define the nodes as follows: [8 marks]

```
public class Node {
    public Object value;
    public Node next;
    public Node(Object v){
        this.value = v;
        this.next = null;
    }
}
```

A class using these Nodes to implement a list would have an instance variable of type Node referring to the list data.

```
public class MLinkedList {
    private Node head;

    // first Node in the list or null if
    // the list is empty
    :
}
```

Complete the definition of method **addToEnd** below of class **MLinkedList** so it adds a new value to the end of the list.

You may not assume there are any additional instance variables in class **MLinkedList**, and you may not add any.

```
public void addToEnd(Object v) {
```

**End of Paper**