Part I. Multiple Choice Questions

For each multiple choice question circle **only one** answer. **Note.** In all questions that refer to singly linked lists or doubly linked lists, methods getNext, getPrevious, setNext and setPrevious are getter and setter methods to obtain and to set the next or previous node in the list.

1. (1 mark) The following statement: `int[] a;` creates an array of integers and each entry of `a` initially stores the value `null`.

   (A) True   √(B) False

2. (2 marks) Consider the following Java statements

   ```
   String s = "hi";
   String t = "hi";
   String s1 = new String("hi");
   String t1 = new String("hi");
   ```

   Which of the following statements has the value `true`?

   ```
   (i) s == t;
   (ii) s1 == t1;
   (iii) s == s1;
   (iv) t == t1;
   ```

   √(A) Only statement (i)   (B) Only statement (ii)   (C) Only (ii) and (iv)   (D) All of them
   (E) None of them

3. (3 marks) Consider the following code fragment

   ```
   int sum = 1;
   int[] arr = null;
   try {
       for (int i = 0; i < 3; ++i) {
           arr[i] = i;
           sum = sum + arr[i];
       }
   }
   catch (ArrayIndexOutOfBoundsException e) {sum = sum + 1;}
   catch (NullPointerException e) {sum = sum - 1;}
   sum = sum + 2;
   ```

   What value does `sum` have at the end of the execution of the above code?

   (A) 1   √(B) 2   (C) 4   (D) 6   (E) 7

4. (3 marks) Consider the following code fragment

   ```
   public class ClassA {
       private static int c = 0;
       public ClassA (int v) {
           c = v;
       }
       public static void main(String[] args) {
           ClassA[] a = new ClassA[3];
           for (int i = 0; i < 3; ++i)
               a[i] = new ClassA(i);
           System.out.println(a[0].c + "," + a[1].c + "," + a[2].c);
       }
   }
   ```

   What is printed when the above code is executed?

   (A) 0,0,0   (B) 0,1,2   (C) 1,2,3   √(D) 2,2,2   (E) 3,3,3

Consider the following three Java classes to answer Questions 5, 6, and 7:

```java
public class ClassA {
    public int sum = 0;
    public void m() {System.out.println("ClassA");}
    public ClassA() {sum = 1; }
}
public class ClassB extends ClassA {
    public void m() {System.out.println("ClassB");}
    public ClassB() {sum = 2; }
}
public class ClassC extends ClassB {
    public void m() {System.out.println("ClassC");}
    public ClassC() {sum = 2; }
}
```

5. (3 marks) Consider now the following code fragment

```java
ClassC var1 = new ClassA(); // Line 1
ClassB var2 = new ClassC(); // Line 2
```

   Which line(s) cause compilation errors?

   √(A) Line 1   (B) Line 2   (C) Lines 1 and 2   (D) None

6. (3 marks) Consider the following code fragment

```java
ClassB var3 = (ClassB) new ClassA();
ClassB var4 = (ClassB) new ClassC();
var3.m();
var4.m();
```

   What does this code fragment print when it is executed?

   (A) "ClassA" and "ClassC"   (B) "ClassB" twice   (C) "ClassA" only
   (D) Nothing. The code has compilation errors.   √(E) The code produces a runtime error.

7. (2 marks) Consider the following code fragment

```java
ClassA var5 = new ClassA();
if (var5.sum == 0) var5 = new ClassC();
else var5 = new ClassB();
var5.m();
```

   What does this fragment print when it is executed?

   (A) "ClassA"   √(B) "ClassB"   (C) "ClassC"

   (D) Nothing. The code has compilation errors.   (E) The code produces a runtime error.

---

8. (2 marks) What is the output produced by the following code: `C var = new C(); var.foo();`?

```java
public class C {
    public int i = 0;
    public void change(int i) {
        i = 2;
    }
    public void foo() {
        i = 4;
        {int i = 7;}
        change(i);
        System.out.println(i);
    }
}
```

   (A) 0   (B) 2   √(C) 4   (D) 7

9. (2 marks) Consider the following code fragment

```
Integer[] arr = new Integer[2];
for (int i = 0; i < 2; ++i)
    arr[i] = new Integer(i);
```

What value is stored in `arr[0]`?

(A) The value 0   √(B) The address of object `Integer(0)` created in the third line
(C) The address of `arr`   (D) The object `Integer(0)` created in the third line

10. (2 marks) Consider the following interface `classA` and Java class `ClassB`

    ```
    public interface ClassA
    public class ClassB implements ClassA
    ```

    Consider the following statements

    (*i*) `ClassA varA = new ClassA();`
    (*ii*) `ClassB varB = new ClassA();`

    Which of these statements is incorrect?

    (A) Statement (*i*)   (B) Statement (*ii*)   √(C) Both statements   (D) None

11. (3 marks) Consider the following doubly linked list of node objects of class `DoubleLinkedNode`. Class `DoubleLinkedNode` has methods `setNext`, `setPrevious`, `getNext` and `getPrevious` to set and to access the next and previous node in the list, respectively. Let `head` be a reference to the first node in the linked list and `p` be a variable referencing the second node in the list.
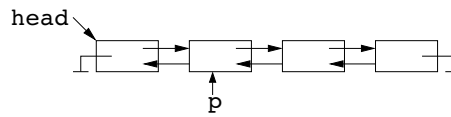
Which code correctly removes only node `p` from the list?

(A) `head = p.getNext(); head.setPrevious(null);`
(B) `p.setNext(null); p.setPrevious(null); p = null;`
√(C) `head.setNext(p.getNext()); (p.getNext()).setPrevious(head);`
(D) `head = head.getNext(); head.setPrevious(null);`
(E) `p.setNext(p.getNext()); p.setPrevious(null); head = p.getNext();`

12. (2 marks) Consider the following code fragment. Each node of class `LinearNode` stores an integer value that can be accessed with method `getValue()`.

```
LinearNode<Integer> aNode = new LinearNode<Integer>(1);
LinearNode<Integer> bNode = new LinearNode<Integer>(2);
LinearNode<Integer> cNode = new LinearNode<Integer>(3);
aNode.setNext(bNode);
bNode.setNext(cNode);
cNode.setNext(aNode);
int sum = 0;
LinearNode<Integer> tmp, curr = aNode;
while (curr != null) {
    sum = sum + curr.getValue();
    tmp = curr.getNext();
    curr.setNext(null);
    curr = tmp;
}
```

What is the value of variable `sum` after the above code is executed?

(A) 1   (B) 5   (C) 6   √(D) 7   (E) The program would never terminate

4

13. (2 marks) What will the value of variable `sum` be if the following code fragment is executed on the stacks `x` and `y` shown below?

```
int sum = 0;
while (!x.isEmpty()) {
    int v = x.pop();
    if (y.peek() < v) v = v - y.pop();
    sum = sum + v;
}
```

5	9	1

x

4	7

y
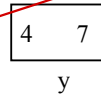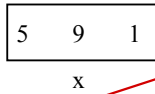
√(A) 4   (B) 7   (C) 8   (D) 10   (E) 15

14. (3 marks) Consider the following code fragment.

```
private int m(char c) {
    return (int) c;
}

public static void main(String[] args) {
    int res = m('a');
    System.out.println(res);
}
```

Which of the following statements regarding the above code fragment is correct?

√(A) The code has compilation errors
(B) The code does not have compilation errors, but it will cause a runtime error
(C) The code has no errors

15. (3 marks) Consider the following code fragment. `Integer` is a wrapper class of Java that represents integer objects. Method `intValue()` of this class returns the `int` value of an `Integer` object. For example, if `Integer intObj = new Integer(3);` the value that `intObj.intValue()` will return is 3.

```
Integer[] arr = new Integer[5];
try {
    int s = 0;
    for (int i = 0; i < 5; ++i) s = s + arr[i].intValue();
}
catch (ArrayIndexOutOfBoundsException e) {
    System.out.println("Invalid index");
}
catch (NullPointerException e) {System.out.println("Null pointer");}
catch (Exception e) {System.out.println("Exception");}
```

What is printed when the above code is executed?

(A) "Invalid index"   √(B) "Null pointer"   (C) "Exception"   (D) Nothing
(E) "Invalid index", "Null pointer" and "Exception"

16. (1 mark) Consider a java class `AnyClass` with constructor `AnyClass()` and the following statement.

```
Object var = new AnyClass();
```

(A) The statement will cause a compilation error
(B) The execution of the statement will cause a runtime error
√(C) The statement does not cause compilation or runtime errors

17. (2 marks) Consider initially empty stack `s` and queue `q` that can store integer values. For the following code fragment

```
int i;
for (i = 0; i < 5; i = i+1) s.push(i);
for (i = 0; i < 2; i = i + 1) q.enqueue(s.pop());
for (i = 0; i < 2; i = i + 1) s.push(q.dequeue());
System.out.println(s.pop());
```

What value is printed?

(A) 2   √(B) 3   (C) 4   (D) 5

18. (4 marks) Consider the following implementation of a stack using an array and two variables `top` and `bottom` which are the indices of the top and bottom of the stack, respectively.

```
public class ArrStack<T> {
    private T[] arr;
    private int top, bottom;
    public ArrStack() {
        arr = (T[]) new Object[10];
        top = 10;   // index of the element at the top of the stack
        bottom = 9; // bottom is at the last index of the array
    }
    public void push(T element) {
        top = top - 1;
        arr[top] = element;
    }
    public T pop() {
        top = top + 1;
        return arr[top + 1];
    }
}
```

Which of the following statements is true?

(A) This is a correct implementation of a stack
(B) This implementation is incorrect because the bottom of the stack must always be at index zero
(C) This implementation is incorrect because the `pop` method can cause a null pointer exception
√(D) This implementation is incorrect because the `push` operation can cause and array index out of bounds exception
(E) This implementation is incorrect because `top` cannot be initialized to 10 as the indices of the array `arr` are 0, 1, ..., 9.

Part II. Written Answers

19. (8 marks) Consider the following code.

public class C {
```
      private static int i = 0;
```
     public C() {i = 0;}

     private void method1 (int i) throws Exception1, Exception2 {
```
          try {
                if (i == 0) throw new Exception1();
                method2 (i);
                this.i = 7;
          } catch (Exception1 e) {
                i = 5;
                System.out.println("Exception 1 caught in method1");
                method2(i);
                if (i == 5) throw new Exception2();
          }
      }
```
     private void method2 (int x) throws Exception1 {
```
          int i = 4;
          try {
                if (x > 4) throw new Exception1();
                else throw new Exception2();
          }
          catch (Exception2 e) {System.out.println("Exception 2 caught in method 2");}
      }
```
     public static void main (String[] args) {
```
          C varc = new C();
          try {
                varc.method1(0);
                i = 10;
          }
          catch (Exception1 e1) {System.out.println("Exception 1 caught in main");}
          catch (Exception2 e) {System.out.println("Exception 2 caught in main");}
          System.out.println("i = " + i);
      }
```
}

`Exception1` and `Exception2` are not parent/child classes of each other. Write all the output produced when this program is executed.

Exception 1 caught in method1

Exception 1 caught in main

i = 0

20. Consider the following code fragment that modifies a singly linked list in which every node stores an integer value. For a node p, p.getValue() returns the value stored in p.

```
public void modify (LinearNode head, int k) {
    LinearNode curr = head, next = curr.getNext();
    while (curr != null)
        if (curr.getValue() == k) {            //Line 1
            curr.setNext(next.getNext()); //Line 2
            next.setNext(curr);            //Line 3
            curr = curr.getNext();         //Line 4
        }
        else {
            curr = next;
            next = curr.getNext();         //Line 5
        }
}
```

Consider the following three linked lists
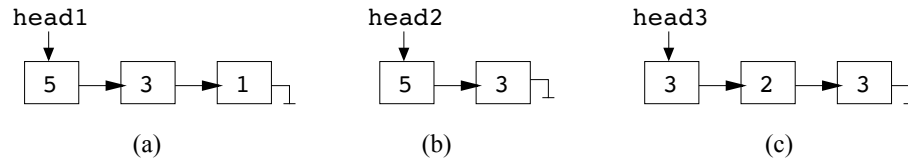
(4 marks) If method `modify(head1,3)` crashes when executed on list (a) of the figure explain what exception is thrown, why it is thrown, and by which line of the code (lines are labeled). If the method does not crash indicate whether the code would terminate or not, and draw the list (must draw all the nodes) after either the algorithm terminates or after the while loop has performed 2 iterations (if algorithm does not terminate).

*The code terminates and produces this list:*
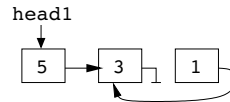
(4 marks) If method `modify(head2,3)` crashes when executed on list (b) of the figure explain what exception is thrown, why it is thrown, and by which line of the code (lines are labeled). If the method does not crash indicate whether the code would terminate or not and draw the list (must draw all the nodes) after either the algorithm terminates or after the while loop has performed 2 iterations (if algorithm does not terminate).

The code throws a null pointer exception when executing Line 2 because next is null.

(5 marks) If method `modify(head3,3)` crashes when executed on list (c) of the figure explain what exception is thrown, why it is thrown, and by which line of the code (lines are labeled). If the method does not crash indicate whether the code would terminate or not and draw the list (must draw all the nodes) after either the algorithm terminates or after the while loop has performed 2 iterations (if algorithm does not terminate).

The code never ends and after two iterations of the loop the list is this:

8
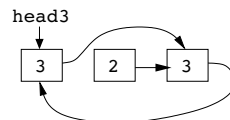
21. (18 marks) Write in Java or in **detailed pseudocode** like the one used in the lecture notes an algorithm `removeAll(head,k)` that receives as parameter a reference `head` to the first node of a singly linked list storing integer values and a value `k` and it removes from the linked list all nodes storing the value `k`. For example, for the singly linked list shown on the left side of the following figure and `k = 5`, the algorithm must remove the nodes storing the value 5, so at the end the list must be as show on the right side of the figure. If no node stores the value `k` then the list must not be modified.
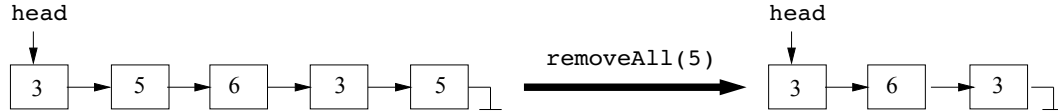
For simplicity, <u>assume that the value k is not stored in the first node of the list</u>. Let p and q be references to nodes of the list. **The ONLY methods** that you can use to manipulate the linked list are the following: p.getValue() returns the value stored in p, p.getNext() returns a reference to the next node in the list after p, p.setNext(q) makes node p point to node q.

You CANNOT use any auxiliary data structures (you cannot use an array, stack, queue, another list, and so on). You **CANNOT** create a second linked list and copy the values from the first list to the second one.

Hint. Use two variables: `prev` and `current`. Scan the list and for each node determine whether it stores the value `k` and if so remove it from the list; which pointer needs to change to remove this node? How do you update `prev` and `current`?

Algorithm removeAll(head,k)
    current = head.getNext()
    prev = head
    while current $\neq$ null **do**
        if current.getValue() $= k$ **then** {
            prev.setNext(current.getNext())
            current = current.getNext()
        }
        else {
         prev = current
         current = current.getNext()
        }

For the following two questions write algorithms in Java or in **very detailed** Java-like pseudocode like the one used in the lecture notes.

23. (18 marks) Given a singly liked list whose first node is referenced by `front`, write an algorithm called `swap(p)` that swaps a given node p with node `p.getNext()`. For example, for the following singly linked list and node p, after swapping nodes p and `p.getNext()` the list should be as in the figure at the bottom.
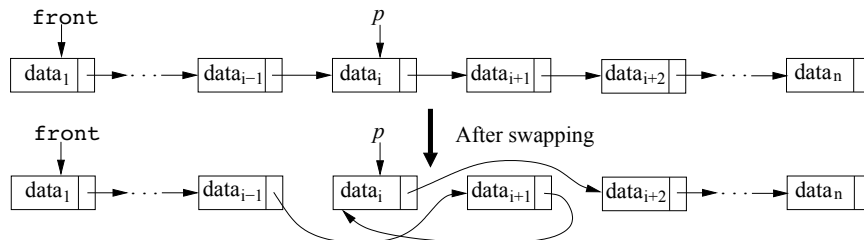
You **cannot** just swap the information stored in nodes p and `p.getNext()`; if you do this you will not get any marks. You must change the pointers, so the nodes are swapped as required.

The **only** methods that you can use in your algorithm to manipulate the linked list are `getNext` and `setNext`. To make the question simpler you might assume that p is a node of the list which is neither the first node or the last node.

Algorithm Swap(p)
In: Node p of a singly linked list; p is neither the first or last node
Out: Nothing, only swap p and p.getNext().
// Find the node that points to p
prev = front
while prev.getNext() ≠ p do
        prev = prev.getNext()
nex = p.getNext()
prev.setNext(nex)
p.setNext(nex.getNext())
nex.setNext(p)

10

24. (18 marks) Consider an array A of length n storing n integer values (the values can be positive, negative, or zero); the same value can appear several times in array A. Write an algorithm $\texttt{noDups}(A, B, n)$ that copies all the different values in A into a second array B of length n (so every value in A must appear in B, but only once; thus, B must contain no duplicated values). For example, if A is the array shown on the left, array B should be as shown on the right.

A	7	5	7	-2	0	5	7	1		B	7	5	-2	0	1			
	0	1	2	3	4	5	6	7			0	1	2	3	4	5	6	7

Your algorithm must receive as parameter arrays A and B and the number n of values stored in A. The algorithm **must** return the number of values copied into B. So, for the above example your algorithm must return the value 5. A and B are arrays of integers so you **cannot** store the value $\texttt{null}$ in an entry of A or B.

You **cannot use made-up array methods** like $\texttt{arrayFind}$, $\texttt{arrayDelete}$, $\texttt{arrayAdd}$, and so on to find, add, delete information of an array. If you do this you will not receive any marks. If you, for example, need to check whether a value x is stored in an array you **must** write the algorithm to do this. The only array operations that your algorithm can use are to read the value stored in some index of an array (like $\cdots = A[index] \cdots$) or to store an integer value in an entry of an array (like $B[index] = \cdots$).

You algorithm must work on **any** array A storing integer values and not only on the array A given in the figure as an example.

```
Algorithm noDups (A, B, n)
In: Array A storing n integer values, empty array B
Out: Number of different values in A, which were copied to B
    numB = 0  // Number of values in B
    for i = 0 to n-1 do {
            // Look for A[i] in B
            j = 0
            while (j < numB) and (B[j] ≠ A[i]) do
                    j = j+1
            if j < numB then {  // A[i] is not in B, add A[i] to B
                    B[numB] = A[i]
                    numB = numB+1
            }
    }
    return numB
```