

Western University
Department of Computer Science
CS1027b Foundations of Computer Science II
Midterm Exam
March 11, 2017

Last Name: _____

First Name: _____

Student Number: _____

Section Number (1 - Hughes, 2 - Solis-Oba): _____

PART I	
PART II	
16	
17	
18	
19	
20	
21	
22	
23	
24	
Total	

Instructions

- Fill in your name, student number, and section.
- The exam is 2 hours long and it has a total of 100 marks.
- The exam has 9 pages and 24 questions.
- The first part of the exam consist of multiple choice questions. For each question circle only **one** answer.
- For the second part of the exam, answer each question only in the space provided.
- When you are done, raise your hand and one of the TA's will collect your exam.

Part I. Multiple Choice Questions

For each multiple choice question circle **only one** answer.

1. (1 mark) Consider the following Java statement

```
Integer i = new Integer(1);
```

What value is stored in i?

- (A) The value 1 ☒ (B) The address of object Integer(1) (C) The object Integer(1)

2. (1 mark) The following statement: `String[] s = new String[10];` creates an array object and each one of the 10 entries of the array stores an empty ("") string.

- (A) True ☒ (B) False

3. (1 mark) Consider the following Java code

```
Integer a = new Integer(1), b = new Integer(1);  
if (a == b) m1();  
else m2();
```

If this code is executed which method will be invoked m1 or m2?

- (A) m1() ☒ (B) m2()

4. (1 mark) Consider the two following Java classes

```
public interface A  
public class B implements A
```

Consider the following statements

- (i) `A varA = new A();`
(ii) `B varB = new B();`

Which of these statements is incorrect?

- ☒ (A) Statement (i) (B) Statement (ii) (C) Both statements (D) None

5. (2 marks) Consider the following Java code

```
private int m(int x) {  
    if (x <= 1) return x;  
    else return 2*m(x-2);  
}
```

Which value does the call `m(6)` return?

- (A) 6 (B) 5 (C) 4 (D) 2 ☒ (E) 0

6. (1 mark) Let class X be a subclass or child class of class Y. Consider the following Java code

```
X varX = new X();  
Y varY = varX; // Line 1  
varY = new Y();  
varX = varY; // Line 2
```

Which line(s) generate compilation error(s)? (Line numbers are indicated in the comments).

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

7. (2 marks) Let class X be a subclass of class Y and let `m()` be a `void public` method in class X not in class Y. Consider the following Java code

```
Y varY = new Y();  
X varX = (X) varY; // Casting  
varX.m();
```

Which kind of error will this fragment produce?

- (A) A compilation error ☒ (B) A runtime error (C) No error

8. (2 marks) Let class X be a subclass of class Y. Let `m()` be a method defined in class X and let `m()` be defined in class Y as well (so method `m()` in class X overrides method `m()` in class Y). Consider the following Java code

```
Y varY = new X();
varY.m();
```

Which of the following statement is correct?

- (A) The above code produces compilation error(s).
(B) The above code produces runtime error(s).
☒ (C) There are no errors and in the second statement, method `m()` from class X is invoked.
(D) There are no errors and in the second statement, method `m()` from class Y is invoked.

9. (2 marks) Consider the following Java code

```
int x = 0, y = 1;
try {
    for (int i = 0; i < 10; i = i+1) y = y + i;
    y = y / x;
    for (int j = 0; j < 5; j = j+1) x = x + 1;
}
catch (Exception e) {System.out.println("Error in program");}
System.out.println(x);
```

Which value is printed for `x`?

- ☒ (A) 0 (B) 4 (C) 5 (D) No value is printed

10. (2 marks) Consider the following Java code

```
int j = 0, k = 0;
Integer i = null;
int[] A = new int[10];
try {
    for (j = 0; j <= 10; j = j+1) {
        A[j] = k;
        k = k+1;
    }
    k = k + i.intValue();
}
catch (ArrayIndexOutOfBoundsException e) {System.out.println("Error");}
catch (NullPointerException e) {k = 1;}
catch (Exception e) {k = 2;}
j = k;
```

Which value does `j` have at the end of the execution of the above program?

- (A) 0 (B) 1 (C) 2 ☒ (D) 10 (E) 11

11. (1 mark) Consider the following Java code

```
private void m() {
    int i = 2;
    Integer j = new Integer(2);
}
```

Which of the following statements is correct?

- (A) `i` and the object referenced by `j` are allocated memory in the execution stack (or runtime stack or call stack).
(B) `i` and the object referenced by `j` are allocated memory in the heap.
☒ (C) `i` is allocated memory in the execution stack and the object referenced by `j` is allocated memory in the heap.

(D) `i` is allocated memory in the heap and the object referenced by `j` is allocated memory in the execution stack.

12. (1 mark) An ADT specifies a set of operations and the way in which they must be implemented.

(A) True ☒ (B) False

13. (2 marks) Consider a stack `S` and a queue `Q` storing integer values. For the following code

```
for (i = 0; i < 5; i = i+1) S.push(i);
for (i = 0; i < 5; i = i + 1) {
    tmp = S.pop();
    Q.enqueue(tmp);
}
tmp = Q.dequeue();
```

What is the value of `tmp` at the end?

☒ (A) 4 (B) 3 (C) 2 (D) 1 (E) 0

14. (1 mark) Consider the following Java code.

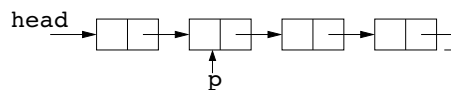
```
public class B {
    private int a;
    public int b;
    public B() {
        a = 1;
        b = 2;
    }
}

public class A {
    private int s;
    public int t;
    public A() {
        B varB = new B();
        int z = s + t; // Line 1
        z = z + varB.a; // Line 2
        z = z + varB.b; // Line 3
    }
}
```

Which line(s) will cause compilation error(s)? (Line numbers are indicated in the comments)

(A) Line 1 ☒ (B) Line 2 (C) Line 3 (D) Lines 1,2 (E) Lines 1,3 (F) Lines 2,3 (G) Lines 1,2,3

15. (1 mark) Consider a linked list of node objects of class `LinearNode`. Assume that the list has at least 2 nodes. Class `LinearNode` has methods `setNext` and `getNext` to set and to access the next 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) `p = null;`
(B) `head = p.getNext();`
(C) `head = head.getNext();`
☒ (D) `head.setNext(p.getNext());`
(E) `p.setNext(p.getNext().getNext());`

Part II. Written Answers

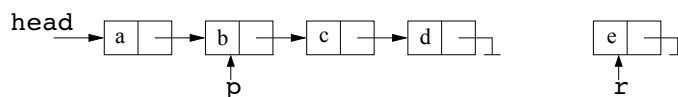
16. (6 marks) Consider the following Java code.

```
public class A {
    private static void m1(int i) throws Exception2 {
        try {
            if (i == 1) throw new Exception2();
            m2();
        } catch (Exception1 e) {System.out.println("m1: Exception 1");}
    }
    private static void m2() throws Exception1 {
        int x = 0;
        try {
            m1(1);
            if (x == 0) throw new Exception1();
        } catch (Exception2 e) {System.out.println("m2: Exception 2");}
    }
    public static void main(String[] args) {
        try {
            m1(2);
        }
        catch (Exception2 e) {System.out.println("main: Exception 2");}
    }
}
```

`Exception1` and `Exception2` are not parent/child classes of each other. Write **all** the output that will be printed by the above algorithm.

m2: Exception2

17. (6 marks) Consider the following linked list, and node `r`



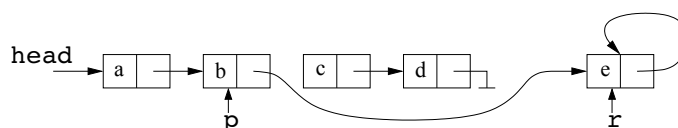
(1 mark) Does the following code correctly insert node `r` in the list after node `p`? (Circle one answer)

(A) Yes ☒ (B) No

```
p.setNext(r);
r.setNext(p.getNext());
```

where `getNext` and `setNext` are a getter and setter methods for the class representing the nodes of the linked list to access and set the next node in the list.

(5marks) Draw the linked list resulting after the above code is executed.

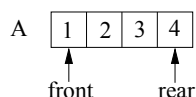


18. (6 marks) Consider a queue of integers implemented using a circular array A of length n . Let **front** and **rear** be the integer indices of the first and last elements of the queue, respectively. The following Java code implements the enqueue operation.

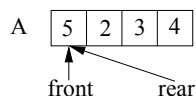
```
private void enqueue (int newValue) {
    rear = (rear + 1) % n;
    A[rear] = newValue;
}
```

Indicate whether this is a correct implementation of the enqueue operation. If the implementation is incorrect, explain why it is incorrect by giving an example showing that the code will not produce the correct result. (**Hint.** Your answer might either be “The algorithm is correct because ...” or it might be “The algorithm is incorrect. Assume a circular array of length x storing y integer values: x_1, x_2, \dots, x_y , where **front** = i_1 and **rear** = i_2 —or just draw the array—. After performing **enqueue(z)** ...”.)

The algorithm is incorrect. Assume a circular array of length 4 storing values 1, 2, 3, 4, where **front** = 0 and **rear** = 3:



After performing **enqueue(5)** the array is



The value 1 has been incorrectly removed.

19. (5 marks) Compute the time complexity of the following algorithm for counting the number of elements in a queue implemented using a circular array of length M storing n elements. You must explain how you computed the time complexity and you must give the order (big-Oh) of the time complexity. (**Hint.** Your answer might be like this: “Number of operations performed outside the **while** loop: x ; number of operations performed in one iteration of the **while** loop: y ; number of iterations of the **while** loop: z ; total number of operations performed by the while loop: w . Total number of operations performed by the algorithm: $x + w$. The order of the time complexity is $O(v)$.”)

```
private int size() {
    int count = 0;
    int i = front;
    while (i != rear) {
        count = count + 1;
        (i = i + 1) % M;
    }
    return count + 1;
}
```

Number of operations outside the while loop is constant; let us denote it with c_1 . Each iteration of the loop performs a constant number, let us denote it c_2 , of operations. The loop is repeated n times, as each iteration the value of i increases by 1 (modulo M) and there are n elements in the queue. Therefore, the total number of operations performed by the while loop is $c_2 \times n$.

The total number of operations performed by the algorithm is $c_1 + c_2 \times n$, which is $O(n)$.

For each one of the following two questions, compute the time complexity of the given code. You must explain how you computed the time complexity and you must give the order (big-Oh) of the time complexity. (See hint in previous question.)

20. (7 marks)

```
int x = 0;
for (int i = 0; i < n; i = i+1)
    if (x < i) x = x+1;
    else i = n + x;
```

Outside the loop a constant number c_1 of operations is performed. Also, each iteration of the for loop performs a constant number c_2 of operations. we now need to determine the number of times that the for loop is repeated:

Since $x = 0$, in the first iteration of the loop the condition of the `if` statement is false, so the value of i is set to n . Therefore, the body of the for loop is repeated only once, and so the total number of operations performed by the loop is $c_1 + c_2$, which is $O(1)$.

21. (7 marks)

```
int x = 1;
for (int i = 0; i < n; i = i+1)
    for (int j = 0; j < n; j = j+1)
        i = i+1;
```

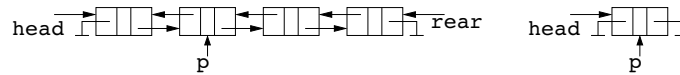
Outside the first loop a constant number c_1 of operations are performed. Now let us consider the inner loop first. Each iteration of the inner loop performs a constant number c_2 of operations. The inner loop is repeated n times for each iteration of the outer loop, as the value of j is initially set to zero, the value of j increases by one in each iteration and the loop ends when $j = n$. Therefore, each execution of the inner loop performs c_2n operations.

Note that each time that the body of the inner loop is executed, the value of i increases by one. Since in the first iteration of the outer loop the value of i is zero and in each iteration of the inner loop the value of i increases by one, then at the end of the first iteration of the outer loop the value of i is equal to n and so the outer loop will exit after just one iteration.

The total number of operations performed by the algorithm is then $c_1 + c_2n$, which is $O(n)$.

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

22. (14 marks) Consider a doubly linked list formed by node objects of class `DoubleNode`. This class provides methods `DoubleNode getPrev()` that returns the previous node in the list, `DoubleNode getNext()` that returns the next node in the list, `setPrev(DoubleNode prev)` to set the previous node in the list, and `setNext(DoubleNode next)` to set the next node in the list. Let `front` be a reference to the first node in the list and `rear` be a reference to the last node in the list. Write an algorithm `remove(DoubleNode p)` that removes node `p` from the linked list. **Note.** You might assume that node `p` is in the list, so the list **has at least one node**.



Solution in Java:

```
private void remove (DoubleNode p) {
    if (p.getPrev() != null)
        (p.getPrev()).setNext(p.getNext());
    else head = p.getNext();
    if (p.getNext() != null)
        (p.getNext()).setPrev(p.getPrev());
    else rear = p.getPrev();
}
```

Solution in pseudocode:

```
Algorithm remove(p)
    if p.getPrev()  $\neq$  null then
        (p.getPrev()).setNext(p.getNext());
    else head = p.getNext();
    if p.getNext()  $\neq$  null then
        (p.getNext()).setPrev(p.getPrev());
    else rear = p.getPrev();
```

23. (14 marks) Consider an array `A` storing `n` integer values. Write a recursive algorithm that receives as parameters `A` and an integer value `n`, and it prints the values of `A` in reverse order in which they appear in `A`. So, for example if `A` is `[7, 1, 6, 9, 2]` and `n = 5`:

7	1	6	9	2
0	1	2	3	4

the algorithm must print out 2 9 6 1 7. (**Hint.** The base case is when $n = 1$; which value should be printed then? In the recursive case, which value is printed before the recursive call?)

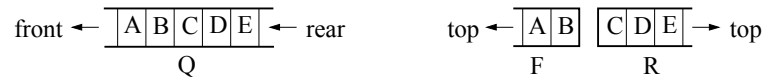
Solution in Java:

```
private void rprint(int[] A, int n) {
    if (n == 1)
        System.out.print(A[0]);
    else {
        System.out.print(A[n-1]+" ");
        rprint(A,n-1);
    }
}
```

Solution in pseudocode:

```
Algorithm rprint(A, n)
    if n = 1 then print the value of A[0]
    else {
        print the value of A[n - 1]
        rprint(A,n-1);
    }
```


24. (14 marks) A queue Q can be implemented with two stacks F and R , as shown in the following figure. An **enqueue** operation is implemented by a **push** operation on R and a **dequeue** operation is implemented with a **pop** operation on F .



However, additional work needs to be performed for the **dequeue** operation when stack F is empty. Write an algorithm to implement the **dequeue** operation using two stacks. You can use stack operations **isEmpty()**, **push(T element)**, and **pop()**. Assume that elements stored in the queue and the stacks are of type T . You can assume that R is not empty.

Solution in Java:

```
private T dequeue() {
    if (!F.isEmpty()) return F.pop();
    else {
        if (R.isEmpty())
            throw new EmptyQueueException();
        while (!R.isEmpty())
            F.push(R.pop());
        return F.pop();
    }
}
```

Solution in pseudocode:

```
Algorithm dequeue() {
    if !F.isEmpty() then return F.pop()
    else
        if R.isEmpty() then ERROR
        else {
            while !R.isEmpty() do
                F.push(R.pop())
            return F.pop()
        }
}
```