CS 1103 Programming 2 - Term 1, 2018-2019

Started on	Wednesday, 26 September 2018, 4:27 PM
State	Finished
Completed on	Wednesday, 26 September 2018, 4:36 PM
Time taken	8 mins 44 secs
Marks	16.67/17.00
Grade	**98.04** out of 100.00

Question **1**

Correct

Mark 1.00 out of 1.00

In the linked list implementation of the queue class, where does the insert method place the new entry on the linked list?

Select one:

- ○ a. At the head.
- ● b. At the tail. ✓
- ○ c. After all other entries that are greater than the new entry.
- ○ d. After all other entries that are smaller than the new entry.

The correct answer is: At the tail.

Question **2**

Correct

Mark 1.00 out of 1.00

How can you drink an entire keg of root beer?

Select one:

- ○ a. (1) take one swallow, then (2) take another swallow.
- ● b. (1) If the keg is empty do nothing, otherwise (2) take one swallow, then drink the rest of the keg. ✓
- ○ c. (1) take one enormous gulp, and (2) wish you hadn't.
- ○ d. (1) drink one keg, and (2) drink another keg.

The correct answer is: (1) If the keg is empty do nothing, otherwise (2) take one swallow, then drink the rest of the keg.

Question **3**

Correct

Mark 1.00 out of 1.00

Consider the tree in the previous question. What is the value stored in the parent node of the node containing 30?

Select one:

- a. 10
- [✓ b. 11] ✓
- c. 14
- d. 40
- e. None of the above

The correct answer is: 11

Question **4**

Correct

Mark 1.00 out of 1.00

How do you study a text book?

Select one:

- a. (1) Read the book on day 1, and (2) read it again each day of the semester.
- [✓ b. (1) If you have reached the end of the book you are done, else (2) study one page, then study the rest of the book.] ✓
- c. (1) Divide the book in two, and (2) study each half.
- d. (1) Cram all the pages in one horrible session, and (2) forget everything the next night.

The correct answer is: (1) If you have reached the end of the book you are done, else (2) study one page, then study the rest of the book.

Question **5**

Correct

Mark 1.00 out of 1.00

Consider the tree in the previous question. What is the order of nodes visited using an in-order traversal?

Select one:

- a. 1 2 3 7 10 11 14 30 40
- [✓ b. 1 2 3 14 7 10 11 40 30] ✓
- c. 1 3 2 7 10 40 30 11 14
- d. 14 2 1 3 11 10 7 30 40

The correct answer is: 1 2 3 14 7 10 11 40 30

Question **6**

Correct

Mark 1.00 out of 1.00

What are two parts to recursion?

Select one:

- [✓ a. (1) If the problem is easy, solve it immediately, and (2) If the problem can't be solved immediately, divide it into smaller problems.] ✓
- b. (1) Divide the problem into smaller problems, and (2) give immediate solutions for the hard problems.
- c. (1) Discard the hard cases , and (2) solve the easy easy cases.
- d. (1) Solve the problem by asking it to solve itself, (2) Solve the easy cases in one step.

The correct answer is: (1) If the problem is easy, solve it immediately, and (2) If the problem can't be solved immediately, divide it into smaller problems.

Question 7

Correct

Mark 1.00 out of 1.00

How many leaves does the tree below have?

```
    14
   / \
  2   11
 / \  / \
1  3 10  30
    /   /
   7  40
```

Select one:

- ○ a. 2
- ● b. 4 ✓
- ○ c. 6
- ○ d. 8
- ○ e. 9

The correct answer is: 4

Question 8

Correct

Mark 1.00 out of 1.00

Suppose cursor refers to a node in a linked list (using the IntNode class with instance variables called data and link). What statement changes cursor so that it refers to the next node?

Select one:

- ○ a. cursor++;
- ○ b. cursor = link;
- ○ c. cursor += link;
- ● d. cursor = cursor.link; ✓

The correct answer is: cursor = cursor.link;

Question 9

Correct

Mark 1.00 out of 1.00

Given the following piece of code:

```
class Student { public void talk(){} }
public class Test{
        public static void main(String args[]){
                Student t = null;
                try {
                      t.talk();
                      } catch(NullPointerException e){
                            System.out.print("There is a NullPointerException. ");
                      } catch(Exception e){
                            System.out.print("There is an Exception. ");
                      }
                      System.out.print("Everything ran fine. ");
        }
}
```

what will be the result?

a. If you run this program, the following is printed:
There is a NullPointerException. Everything ran fine.

b. If you run this program, the following is printed:
There is a NullPointerException.

c. If you run this program, the following is printed:
There is a NullPointerException. There is an Exception.

d. This code will not compile, because in Java there are no pointers.

Select one:

● a. ✓

○ b.

○ c.

○ d.

The correct answer is: a.

Question **10**

Correct

Mark 1.00 out of 1.00

For a linked list to be used in a program, that program needs:

i. A variable that refers to the first node in the list.
ii. A pointer to the first node.
iii. A null pointer in the last node.

Select one:

- ○ a. i and ii
- ○ b. i
- ○ c. ii and iii
- ● d. i, ii and iii ✓

The correct answer is: i, ii and iii

Question **11**

Correct

Mark 1.00 out of 1.00

Which statements are correct regarding Java's predefined class called *Throwable*?

Select one or more:

- [✓] a. The class Throwable represents all possible objects that can be thrown by a throw statement and caught by a catch clause in a try...catch statement. ✓
- [✓] b. The thrown object must belong to the class Throwable or to one of its (many) subclasses such as Exception and RuntimeException. ✓
- [✓] c. The object carries information about an exception from the point where the exception occurs to the point where it is caught and handled. ✓
- [ ] d. A Throwable contains a snapshot of the execution stack of its thread at the time it was called.

The correct answers are: The class Throwable represents all possible objects that can be thrown by a throw statement and caught by a catch clause in a try...catch statement., The thrown object must belong to the class Throwable or to one of its (many) subclasses such as Exception and RuntimeException., The object carries information about an exception from the point where the exception occurs to the point where it is caught and handled.

Question **12**

Correct

Mark 1.00 out of 1.00

The following code writes out the name of a day of the week depending on the value of *day*. True or False?

```
String dayName = null;
switch (day) {
case 1:
dayName = "Sunday";
break;
case 2:
dayName = "Monday";
break;
case 3:
dayName = "Tuesday";
break;
case 4:
dayName = "Wednesday";
break;
case 5:
dayName = "Thursday";
break;
case 6:
dayName = "Friday";
break;
case 7:
dayName = "Saturday";
break;
}
System.out.println(dayName);
```

Select one:

[✓ True] ✓

False

The correct answer is 'True'.

Question **13**

Correct

Mark 1.00 out of 1.00

In the following method, what is the base case?

```
static int xMethod(int n) {
   if (n == 1)
     return 1;
   else
     return n + xMethod(n - 1);
}
```

Select one:

- [● a. n is 1 ✓]
- ○ b. n is greater than 1.
- ○ c. n is less than 1.
- ○ d. no base case.

The correct answer is: n is 1

Question **14**

Partially correct

Mark 0.67 out of 1.00

“Subclasses of the class *Exception* which are not subclasses of RuntimeException require mandatory exception handling.” What are the practical implications of this statement?

Select one or more:

- ☐ a. If a method can throw such an exception, then it must declare this fact by adding a throws clause to the method heading.
- [✓ b. If a routine includes any code that can generate such an exception, then the routine must deal with the exception.] ✓
- ☐ c. The routine cannot handle the exception by adding a throws clause to the method definition.
- [✓ d. The routine can handle the exception by including the code in a try statement that has a catch clause to handle the exception.] ✓

The correct answers are: If a method can throw such an exception, then it must declare this fact by adding a throws clause to the method heading., If a routine includes any code that can generate such an exception, then the routine must deal with the exception., The routine can handle the exception by including the code in a try statement that has a catch clause to handle the exception.

Question 15

Correct

Mark 1.00 out of 1.00

Consider the tree below. What is the order of nodes visited using a pre-order traversal?

```
    14
   /  \
  2    11
 / \   / \
1  3  10 30
      /  /
     7   40
```

Select one:

a. 1 2 3 7 10 11 14 30 40

b. 1 2 3 14 7 10 11 40 30

c. 1 3 2 7 10 40 30 11 14

[✓ d. 14 2 1 3 11 10 7 30 40] ✓

The correct answer is: 14 2 1 3 11 10 7 30 40

Question **16**

Correct

Mark 1.00 out of 1.00

Study the following three pieces of code. Comments have been removed intentionally.
Can you guess what each does?

(i)

```
public class ProcForInts {
  private int[] items = new int[10];

  private int top = 0;

  /**
   * Procedure
   */
  public void push( int N ) {
     if (top == items.length) {
        int[] newArray = new int[ 2*items.length ];
        System.arraycopy(items, 0, newArray, 0, items.length);
        items = newArray;
     }
     items[top] = N;
     top++;
  }

  /**
   * Procedure
   */
  public int pop() {
     if ( top == 0 )
       throw new IllegalStateException("Can't...");
     int topItem = items[top - 1]
     top--;
     return topItem;
  }

  /**
   * Procedure
   */
  public boolean isEmpty() {
    return (top == 0);
  }
}
```

(ii)

```
public class ProcForInts {
  /**
   * Procedure
   */
  private static class Node {
    int item;
    Node next;
  }
  private Node head = null;

  private Node tail = null;
  /**
   * Procedure
   */
  public void enqueue( int N ) {
    Node newTail = new Node();
    newTail.item = N;
    if (head == null) {
      head = newTail;
      tail = newTail;
    }
    else {
      tail.next = newTail;
      tail = newTail;
    }
  }

  /**
   * Procedure
   */
  public int dequeue() {
    if ( head == null)
       throw new IllegalStateException("Can't...");
    int firstItem = head.item;
    head = head.next;
    if (head == null) {
        tail = null;
    }
    return firstItem;
  }

  /**
```

```
     * Procedure
     */
    boolean isEmpty() {
        return (head == null);
    }

}
```

(iii)

```
public class ProcForInts {
    private static class Node {
        int item;
        Node next;
    }

    private Node top;

    /**
     * Procedure
     */
    public void push( int N ) {
        Node newTop;
        newTop = new Node();
        newTop.item = N;
        newTop.next = top;
        top = newTop;
    }

    /**
     * Procedure
     */
    public int pop() {
        if ( top == null )
            throw new IllegalStateException("Cannot...");
        int topItem = top.item;
        top = top.next;
        return topItem;
    }

    /**
     * Procedure
     */
```

```
  public boolean isEmpty() {
    return (top == null);
  }
}
```

Select one:

- a. (i) is a linked list implementation of a stack; (ii) is an array implementation of a stack; (iii) is a queue
- b. (i) is an array implementation of a stack; (ii) is a linked list implementation of a stack; (iii) is a queue
- c. (i) is a queue; (ii) is a linked list implementation of a stack; (iii) is an array implementation of a stack
- d. (i) is an array implementation of a queue; (ii) is a linked list implementation of a queue; (iii) is a stack
- [✓ e. (i) is an array implementation of a stack; (ii) is a queue; (iii) is a linked list implementation of a stack] ✓

The correct answer is: (i) is an array implementation of a stack; (ii) is a queue; (iii) is a linked list implementation of a stack

Question **17**

Correct

Mark 1.00 out of 1.00

Which of the following statements are true?

Select one:

- [✓ a. The Fibonacci series begins with 0 and 1, and each subsequent number is the sum of the preceding two numbers in the series.] ✓
- b. The Fibonacci series begins with 1 and 1, and each subsequent number is the sum of the preceding two numbers in the series.
- c. The Fibonacci series begins with 1 and 2, and each subsequent number is the sum of the preceding two numbers in the series.
- d. The Fibonacci series begins with 2 and 3, and each subsequent number is the sum of the preceding two numbers in the series.

The correct answer is: The Fibonacci series begins with 0 and 1, and each subsequent number is the sum of the preceding two numbers in the series.