Version 1

Instructions

- **Write your name and version number on the top of the yellow paper.**
- **Answer all questions on the yellow paper.**
- **One question per page.**
- **Use only one side of the yellow paper.**

1. (16 Points) Multiple Choice:

A. (2 Points) In the ADT list, when an item is deleted from position i of the list, _____.

a. the position of all items is decreased by 1
b. the position of each item that was at a position smaller than i is decreased by 1
c. the position of each item that was at a position greater than i is decreased by 1
d. the position of each item that was at a position smaller than i is increased by 1 while the position of each item that was at a position greater than i is decreased by 1

B. (2 Points) A(n) _____ is a Java construct that enables a programmer to define a new data type.

a. class
b. method
c. data field
d. object

C. (2 Points) If you attempt to use a reference variable before it is instantiated, a(n) _____ will be thrown.

a. IndexOutOfBoundsException
b. InstantiationException
c. IllegalAccessException
d. NullPointerException

D. (2 Points) A linked list contains components, called _____, which are linked to one another.

a. nodes
b. arrays
c. vectors
d. references

E. (2 Points) Which of the following statements deletes the node that curr references?

```
a. prev.setNext(curr);
b. curr.setNext(prev);
c. curr.setNext(curr.getNext());
d. prev.setNext(curr.getNext());
```

F. (2 Points) If the array: {6, 2, 7, 13, 5, 4} is added to a stack, in the order given, which number will be the first number to be removed from the stack?

a. 6
b. 2
c. 5
d. 4

G. (2 Points) The _____ method of the ADT stack retrieves and then removes the top of the stack.

a. createStack
b. push
c. pop
d. peek

H. (2 Points) A superclass method can be accessed by a subclass, even though it has been overridden by the subclass, by using the _____ reference.

a. super
b. final
c. static
d. new

Version 1

2. (20 Points) Given the following generic MyArray class that contains 10 syntax and logical errors:

```
public class MyArray {
   private Object array = new Object[100];
   private int currentLocation = 0;

   public int size() {
      return currentLocation;
   }

   public boolean isEmpty() {
      return (currentLocation <= 0);
   }

   public void addElement(I element) {
      array[currentLocation] = element;
   }

   public void getElement(int index) {
      I element = null;
      if ((index < currentLocation) || (index >= 0)) {
         element = (I) array[index];
      }
      return element;
   }

   public void replaceElement(I newElement, int index) {
      if ((index >= currentLocation) && (index < 0)) {
         System.out.println("Error");
      }
      array[index] = newElement;
   }

   public void removeElement(int index) {
      if ((index <= currentLocation) || (index < 0)) {
         System.out.println("Error");
      }
      for (int i = index - 1; i < currentLocation; i++) {
         array[i - 1] = array[i];
      }
      array[currentLocation] = null;
   }

   public void clear() {
      for (int i = 0; i < array.length; i++) {
         array[i] = null;
      }
      currentLocation = 0;
   }
}
```

Re-write the MyArray class and fix the 10 syntax and logical errors.

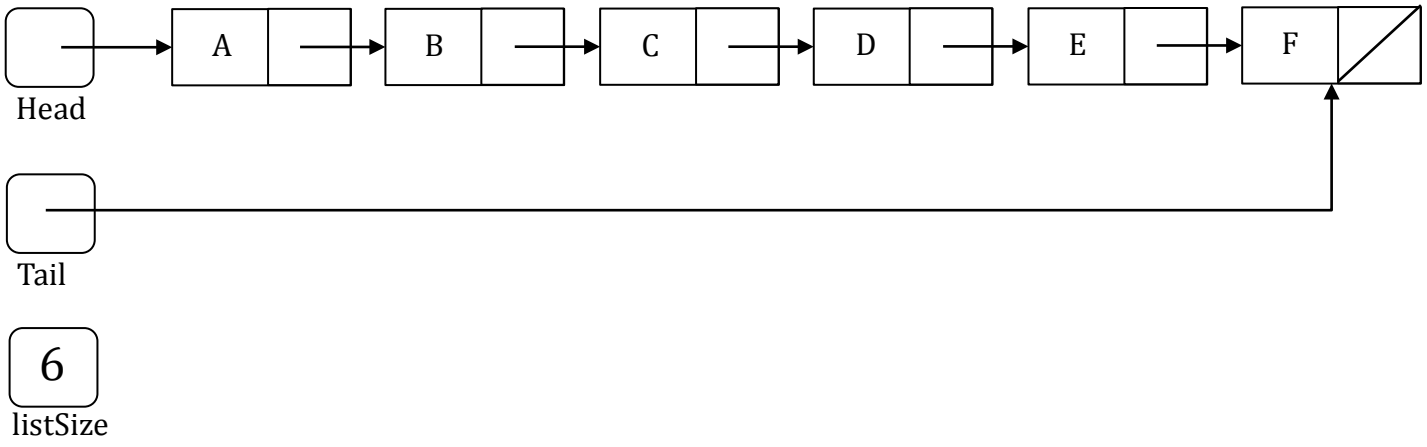
Version 1

3. (40 Points) Given the following definition for **QueueInterface<I>**

```
import java.util.Vector;

public interface QueueInterface<I> {
   // returns true if Queue is empty
   // returns false otherwise
   public boolean isEmpty();

   // returns the size of the Queue
   public int size();

   // adds the specified element
   // to the Queue
   // trying to enqueue into a full array should return without queueing anything
   public void enqueue(I element);

   // removes and returns the front
   // of the Queue
   public I dequeue();

   // returns a Vector containing all the
   // elements in the Queue
   public Vector<I> peekAll();

   // tests if this Queue is equal to the
   // Queue specified by obj
   // Two Queues are equal if they have
   // the same size and all their elements
   // are equal
   public boolean equals(Object obj);
}
```

Write the complete Java class for the **ArrayBasedQueue** that implements the given **QueueInterface<I>**.

4. (30 Points) Given the following list:



And the following method:

```
public void doStuff1() {
   Node[] nodes = new Node[listSize];

   Node node = head;
   int i = 0;
   while (node != null) {
      nodes[i++] = node;
      node = node.getNext();
   }

   for ( i = 0 ; i < listSize ; i += 2 ) {
      nodes[i+1].setNext(nodes[i]);
   }

   for ( i = (listSize - 1) ; i > 1 ; i -= 2 ) {
      nodes[i-3].setNext(nodes[i]);
   }

   head = nodes[1];
   tail = nodes[listSize - 2];
   nodes[listSize-2].setNext(null);
}
```

Draw the list after `doStuff1()` has finished executing.

CMP-338 106 Points Total Midterm Spring 2018

Version 2

Instructions

- **Write your name and version number on the top of the yellow paper.**
- **Answer all questions on the yellow paper.**
- **One question per page.**
- **Use only one side of the yellow paper.**

1. (16 Points) Multiple Choice:

A. (2 Points) In the ADT list, when an item is inserted into position i of the list, _____.

a. the position of all items is increased by 1
b. the position of each item that was at a position smaller than i is increased by 1
c. the position of each item that was at a position greater than i is increased by 1
d. the position of each item that was at a position smaller than i is decreased by 1 while the position of each item that was at a position greater than i is increased by 1

B. (2 Points) A(n) _____ is an instance of a class.

a. method
b. data field
c. interface
d. object

C. (2 Points) When you declare a variable that refers to an object of a given class, you are creating a(n) _____ to the object.

a. interface
b. reference
c. method
d. ADT

D. (2 Points) The last node of a linear linked list _____.

A. has the value null
B. has a next reference whose value is null
C. has a next reference which references the first node of the list
D. cannot store any data

E. (2 Points) I Which of the following statements deletes the first node of a linear linked list that has 10 nodes?

```
a. head.setNext(curr.getNext());
b. prev.setNext(curr.getNext());
c. head = head.getNext();
d. head = null;
```

F. (2 Points) If the array: {6, 21, 35, 3, 6, 2, 13} is added to a stack, in the order given, which of the following is the top of the stack?

a. 2
b. 6
c. 3
d. 13
e. 35

G. (2 Points) The _____ method of the ADT stack retrieves the top of the stack, but does not change the stack.

a. createStack
b. push
c. pop
d. peek

H. (2 Points) The constructor of a subclass can call the constructor of the superclass by using the _____ reference.

a. extends
b. new
c. super
d. import

Version 2

2. (20 Points) Given the following generic MyArray class that contains syntax and logical errors:

```
public class MyArray {
   private I[] array = new Object[100];
   private int currentLocation = 0;

   public int size() {
      return currentLocation;
   }

   public boolean isEmpty() {
      return (currentLocation = 0);
   }

   public void addElement(I element) {
      array[currentLocation--] = element;
   }

   public void getElement(int index) {
      I element = null;
      if ((index >= currentLocation) && (index >= 0)) {
         element = (I) array[index];
      }
   }

   public void replaceElement(I newElement, int index) {
      if ((index >= currentLocation) && (index < 0)) {
         System.out.println("Error");
      }
      array[index] = newElement;
   }

   public void removeElement(int index) {
      if ((index < currentLocation) || (index < 0)) {
         System.out.println("Error");
      }
      for (int i = index + 1; i < currentLocation; i++) {
         array[i + 1] = array[i];
      }
      array[--currentLocation] = null;
   }

   public void clear() {
      for (int i = 0; i < array.length; i++) {
         array[i] = null;
      }
      currentLocation = 0;
   }
}
```

Re-write the MyArray class and fix the 10 syntax and logical errors.

Version 2

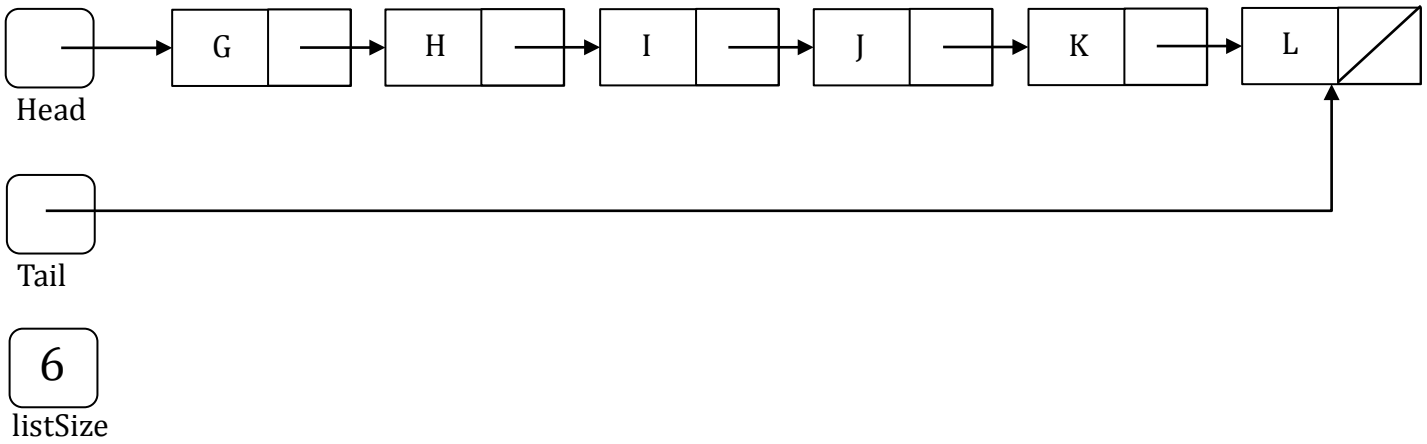
3. (40 Points) Given the following generic definition for **StackInterface<I>**

```
import java.util.Vector;

public interface StackInterface<I> {
   // returns true if Stack is empty
   // returns false otherwise
   public boolean isEmpty();

   // returns the size of the Stack
   public int size();

   // pushes the specified element
   // onto the stack
   // trying to push onto a full array should return without stacking anything
   public void push(I element);

   // pops and returns the element
   // at the top of the stack
   public I pop();

   // returns a Vector containing all the
   // elements in the Stack
   public Vector<I> peekAll();

   // tests if Stack is equal to the
   // Stack specified by obj
   // Two Stacks are equal if they have
   // the same size and all their elements
   // are equal
   public boolean equals(Object obj);
}
```

Write the complete Java class for the **ArrayBasedStack** that implements the given **StackInterface<I>**.

Version 2

4. (30 Points) Given the following list:



And the following method:

```
public void doStuff2() {
   Node[] nodes = new Node[listSize];

   Node node = head;
   int i = 0;
   while (node != null) {
      nodes[i++] = node;
      node = node.getNext();
   }

   for ( i = 1 ; i < listSize-1 ; i += 2 ) {
      nodes[i+1].setNext(nodes[i]);
   }

   for ( i = (listSize - 2) ; i > 2 ; i -= 2 ) {
      nodes[i-3].setNext(nodes[i]);
   }

   nodes[listSize-1].setNext(nodes[2]);
   nodes[listSize-3].setNext(nodes[0]);

   head = nodes[listSize-1];
   tail = nodes[0];
   tail.setNext(null);
}
```

Draw the list after `doStuff2()` has finished executing.