CMSC132 Summer 2017 Midterm 1

First Name (PRINT): ____

Last Name (PRINT): ____

I pledge on my honor that I have not given or received any unauthorized assistance on this examination.

Your signature: ____

Instructions

- This exam is a closed-book and closed-notes exam.
- Total point value is 100 points.
- The exam is a 80 minutes exam.
- Please use a pencil to complete the exam.
- **WRITE NEATLY**. If we cannot understand your answer, we will not grade it (i.e., 0 credit).

1. T/F	/ 10
2. Multiple Choice	/ 25
3. Short Answer	/ 25
4. Programming	/ 40
Total	/100

1. Circle True or False for the following questions (1 pts each)

A. True False The cost of finding a node from a sorted linked list is O(n).

B. True False The “is-a” relationship between classes is called inheritance.

C. True False This will correctly compute the sum of a nonempty list, in which **first** is the first node in the list

```
int sum = 0;
for (Node cur = first; cur != null; cur = cur.next) {
     sum = sum + cur.data;
}
```

D. True False Abstract classes can have constructors.

E. True False If there is a tail reference, which references the last node in the linked list, then appending a new node to the end of a linked list costs constant time.

F. True False In Java, a class can inherit any other class.

G. True False Multiple classes can implement the same interface.

H. True False Items must be comparable to be inserted into a sorted list.

I. True False **super()** must be the first statement of the constructor.

J. True False Aliasing occurs when two or more references to an object exist within a running program.

2. Fill in the blanks/ Multiple Choice

A. (1 pts) In Java, the actual method executed is determined by the type of the object and not the type of the reference. This is called ______________________.

B. (2 pts) When you traverse a linked list starting from the first node in the list using a reference **current**, you know you reached the end of the list if

```
a. current.next == null
b. cur.hasNext() == false
c. current == first
d. current.length == N
```

C. (2 pts) What is the output of the following program

```
class Base {
      final public void show() {
            println("Base");
      }
}
class Derived extends Base {
      public void show() {
            println("Derived");
      }
}
class Main {
      public static void(String[] args){
            Base b = new Derived();
            b.show();
      }
}
```

a) Base
b) Derived
c) Compiler Error
d) Runtime Error

D. (2 pts) The class "Parent" and its inherited class "Child" both implement a method "foo()", printing "parent" and "child" respectively. Which of the 4 choices below reflects the correct output of the following program:

```
Parent v1 = new Parent();
Parent v2 = new Child();
Child v3 = new Child();
System.out.println(v1.foo()+" "+ v2.foo()+" "+ v3.foo());
```

a. parent child child
b. parent parent parent
c. parent child parent
d. child parent child

E. (1 pts) A method defined in a superclass is redefined in a subclass with an identical method signature is called___________.

a. Dynamic binding
b. Method overloading
c. Method overriding
d. Late binding

F. (4 pts) Suppose class C extends class B, and class B implements interface A. Now suppose class D implements interface A, and class E extends class B.

For each of the following operations, circle valid or invalid.

1) C test = new B(); valid invalid
2) C test = new E(); valid invalid
3) B test = new E(); valid invalid
4) A test = new C(); valid invalid

G. (7 pts) Two classes Parent and Child are as follows:

```
Abstract class Parent{
     void foo(){
        System.out.println("parent");
     }
}
class Child extends Parent
{
     void foo(int x){
        System.out.println("child");
     }
}
```

Check the following if each statement is valid or invalid. Circle your answer

```
Parent p1 = new Parent();
```
valid invalid

```
Parent p2 = new Child();
```
valid invalid

```
Child c = new Child();
```
valid invalid

```
p2.foo();
```
valid invalid

```
p1.foo(5);
```
valid invalid

```
c.foo();
```
valid invalid

```
c.foo(10);
```
valid invalid

H. (2 pts)

```
public class Bag implements Comparable<Bag> {
  ...
}
```

Based on the above class definition, which is the signature for the compareTo() method that needs to be implemented by the Bag class?

a) `private int compareTo(Bag other);`
b) `public int compareTo(Bag other);`
c) `public int compareTo(Object other);`
d) `public boolean compareTo(Bag other);`

I.(1 pts) What are the methods you must have if your class implements Iterable<String> ?

J. (2 pts) What are the methods you must have if your class implements Iterator<String>?

K. (1 pts) Suppose a class has a static variable s, and a non-static variable n. It also has static method m1(), and a non-static method m2(). Which of these is true?

a) m1 can access s and n.
b) m2() can access s and n.
c) m2() can access n but not s.
d) m1() can access n only if n is a private variable.

3. Short Answer Questions

A. (3 pts) what is the difference between **`equals`** and **`==`**

B. (3 pts) In a doubly linked list, **`head`** references the first node in the list, **`t`** references a new node. Write code to insert t as the first node.

C. (3 pts) What is the output of the following program

```
public class Test{
      public String name;
      public static int score;
      public static void main(String[] args){
            Test t1 = new Test();
            Test t2 = new Test();
            t1.name = "Java";
            t1.score = 95;
            t2.name = "C++";
            t2.score = 80;
            System.out.println(t1.name);
            System.out.println(t2.name);
            System.out.println(t1.score);
            System.out.println(t2.score);
      }
}
```

D. (3 pts) If a list has more than one element, then what does the following code will do?

```
Node n = head;
while (n.next.next != null) {
    n = n.next;
}
n.next = null;
```

E. (3 pts) We want to insert node t immediately after x. Is this code correct? If it is not correct, explain why?

```
x.next = t;
t.next = x.next;
```

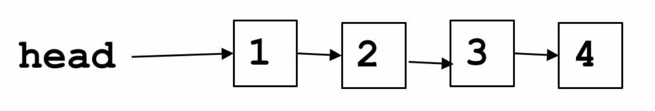
F. (3 pts) Following program has an error. Correct the error.

```
Public class Animal{
      public String name;
      public int age;
}
public class Main{
      public static void main(String[] args) {
           Animal animal;
           animal.name=”tiger”;
           animal.age = 3;
           System.out.println("Name: " + animal.name);
      }
}
```

G. (4 pts) Compare the efficiency of insert and contains methods in array based bag and array based sorted bag

----------------	Insert	Contains
Array Bag	O( )	O( )
Sorted Bag	O( )	O( )

H. (3 pts) Assume linked list



Function **`foo`** is defined as follows:

```
void foo(Node head){
  if(head == null) return;
      print(head.data);
      foo(head.next);
      print(head.data);
}
```

What is the output of **`foo(head)`**

4. Programming Questions

A. (10 pts) Write a method public **`Node array2List(String[] array)`** that takes a string array and returns a LinkedList representation of it. The Node class is given as:

```
public class Node{
      public String data;
      public Node next;
      public Node(String s){data = s;}
}
```

```
public Node array2LinkedList(String[] array) {
        }
```

B. (6 pts) Write a function "**`int length(Node head)`**", which calculates the length of the linked list starting from head. The Node is defined as:

```
public class Node{
      Public String data;
      Public Node next;
      public Node(String s){data = s;}
}
```

Example:

Assume first-->1-->10-->5-->9, tnen **`length(first) returns 4`**

```
int length(Node head){













}
```

C. (10 pts) Given a Node class defined in 4-A, and a Node head that references the first item in the list (or null if empty), write a function **`void insert(Node head, String e)`**, which inserts a string into a correct location in a alphabetically-sorted linked list.

```
void insert(Node h, String e){
















}
```

D. (8 pts) Write a method **Bag<E> copy()**, which returns deep-copied clone of a the array based bag. Assume Bag is iterable and elements in the bag are cloneable. Here is the Bag class:

```
public class Bag<E> implements Iterable<E>{
  protected E[] items;
  protected int N;//number of items in the bag
  protected int capacity = 10;
  public Bag(){
     items = (E[])new Object[capacity];
  }
}
Bag<E> copy (){
```

E. (6 pts) Write a function **Node<E> deleteLast(),** which deletes and returns the last node in the linked list. It returns null if the list is empty.

```
Node<E> deleteLast(){




}
```