CMSC132 Summer 2018 Midterm 1

Solution

First Name (PRINT): ______________________________

Last Name (PRINT): ______________________________

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	/ 15
2. Multiple Choice	/ 20
3. Short Answer	/ 25
4. Programming	/ 40
Total	/100

True/False [15 pts] [1 point each]

T / [✓ F]	If the linked list is sorted, we can use the binary search algorithm to find a key.
T / [✓ F]	A class in Java can have multiple parents
T / [✓ F]	You can instantiate interfaces and abstract classes
[✓ T] / F	If a class implements the Comparable interface, it is guaranteed to have a compareTo() method
[✓ T] / F	static blocks are executed before constructors
[✓ T] / F	If obj1 is a reference copy of obj2, then obj1 == obj2
T / [✓ F]	When an instance field is not given an access modifier, it is private by default
T / [✓ F]	You can always downcast a parent class reference to a child class
T / [✓ F]	Overridden methods must call the parent method with super as the first line
[✓ T] / F	Inner classes can access the private fields of outer classes
[✓ T] / F	Abstract class can define both abstract methods and non-abstract methods
[✓ T] / F	An abstract class can have both abstract and non-abstract child classes.
T / [✓ F]	An abstract method can be defined in a non-abstract class.
T / [✓ F]	A child class can extend a parent or implement an interface, but not do both.
[✓ T] / F	An interface can extend another interface

Fill in the Blanks and Multiple Choice [20 pts]

1.[2 pts] ______________________means the state of an object cannot be changed while ________________ means that it can.

[✓ A. Immutability, mutability]
B. Static, instance
C. Rigidity, flexibility
D. None of the above

2. [2 pts] The final keyword can be used to (Circle all that apply):

[✓ A. To prevent the extension of the class]
[✓ B. To prevent method overriding]
[✓ C. To define symbolic constants]
D. To prevent method overloading

3. [1 pts] Overriding an instance variable is called ___________, because it makes the parent instance variables of the parent class inaccessible

A. Encapsulation
B. Polymorphism
[✓ C. Shadowing]
D. Autoboxing

4. [2 pts] A circular linked list is one in which the last node is connected to the first node. If we set a node curr equal to first node at the start of traversal, the condition to check if we have reached the end of the list then becomes

A. `curr == first`
B. `curr.next ==null`
C. `curr.next.next == first`
[✓ D. `curr.next == first`]

5. [3 pts] Suppose Book is an interface and classes Dictionary and Encyclopedia both implement Book.
Circle whether each statement is valid or invalid.

`Book b = new Book();`	Valid	[✓ Invalid]
`Book d = new Dictionary();`	[✓ Valid]	Invalid
`Encyclopedia e = new Book();`	Valid	[✓ Invalid]

6. [2 pts] What is the output of the following code?

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

```
         A. Base::show() called
         B. Derived::show() called
         C. Compiler Error
         D. Runtime Error
```

7. [2 pts] Assume we compare Rectangles by their areas, implement compareTo method for the rectangle class.

```
Class Rectangle implements Comparable<Rectangle>{
      Private int width, height;
      ...
      Public int compareTo(Rectangle r){
            return (width * height) - (r.width * r.height);





      }

}
```

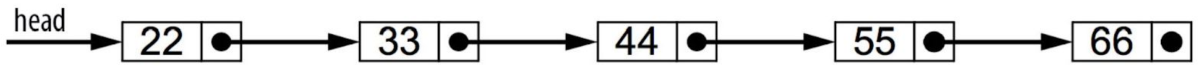
8. [2 pts] What is the output of the following code?

```
class Base {
      public static void show() {
            System.out.println("Base::show() called");
      }
}

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

```
         A. Base::show() called
         B. Derived::show() called
         C. Compiler Error
         D. Runtime Error
```

9.[2 pts] Given print method and a linked list

```
void print(Node n) {
      while (n != null) {
            System.out.print(n.data + “, “);
            n = n.next;
      }
}
```



What will the output be of the code:

```
      head = head.next.next;
      print(head);
```

```
            A. 22, 33, 44,
            B. 33, 44, 55,
            C. 44, 55, 66,
            D. 22, 33, 44,55,66
```

10. [2 pts] What is the output of the following code?

```
class Base {
      public void Print() { System.out.println("Base");}
}
class Derived extends Base {
      public void Print() {  System.out.println("Derived"); }
}
class Main{
      public static void DoPrint( Base o ) { o.Print();  }
      public static void main(String[] args) {
            Base x = new Base();
            Base y = new Derived();
            Derived z = new Derived();
            DoPrint(x);   DoPrint(y);    DoPrint(z);
      }
}
```

A.Base Derived Derived	B. Base Base Derived
C. Base Derived Base	D. Compiler Error

Short Answer [25 pts]

1. [3 pts] Following code is the content of C.java. Does it compile correctly? Explain why?

```
interface A{
  default String m1(){return "A";}
}
interface B{
  default String m1(){return "B";}
}
public class C implements A,B{
  public String m1(){return "C";}
}
```

Yes it compiles. There is no issue with java not knowing which m1 to call between A and B because both methods are overwritten in class C.

2. [3 pts] Following code is the content of C.java. Does it compile correctly? Explain why?

```
interface A{
  String m1();
}
interface B{
  String m1();
}
public class C implements A,B{
  public String m1(){return "C";}
}
```

Yes it compiles. Because neither A nor B provided a default implementation, class C must provide an implementation which is the one used when m1() is called.

3. [3 pts] Following code is the content of C.java. Does it compile correctly? Explain why?

```
interface A{
  public String m1(){}
}

public class C implements A{
  public String m1(){return "C";}
}
```

```
No it does not compile.  Because m1 is not given the default keyword,
it should not have a method body.  There is also a missing semicolon.
```

4. [3 pts] Does the following code compile correctly? Why?

```
Public class A{
      private static int n;
      public int get(){
            return n;
      }
}
```

```
Yes it compiles.  Instance methods are able to access static class
variables.
```

5. [3 pts] What is the difference between abstract classes and interfaces?

```
Abstract classes can have constructors, Interfaces cannot.  Abstract
classes are extended, Interfaces are implemented.  A class can only
extend one abstract class, but can implement multiple interfaces.
etc.
```

6. [3 pts] What is the output of the following code?

```
public class A{
      static{
            System.out.print("A");
      }
      {
            System.out.print("B");
      }
      public static void main(String[] s){
            new A();
            new A();
      }
}
```

ABB

7. [3 pts] What is the output of the following code?

```
public class A{
      static{
            System.out.print("A");
      }
      public A(){
            System.out.print("B");
      }
      public static void main(String[] s){
            new A();
            new A();
      }
}
```

ABB

8. [2 pts] What is the output of the following code?

```
class A{
  private void m1(){System.out.println("A");}
  public void m2(){m1();}
}
public class B extends A{
  private void m1(){System.out.println("B");}
  public static void main(String[] s){
      A a = new B();
      a.m2();
  }
}
```

A

9. [2 pts] Can we apply the binary search algorithm on the unsorted array? Explain Why or why not.

No, because binary search requires us to look for the midpoint of the array and then search half based on whether the element we are looking for is contained in the larger or smaller half. If the array is unsorted, we won't know to look in the larger or smaller half of the array.

Programming [40 pts]

1.[10 pts] In the array based bag class, we increased the capacity when the array is full. User removed some items from the bag, and now we want to shrink the array. Write a method "`private void shrink()`", which shrinks the array by half when the number of items is less than or equal to ¼ of the capacity.

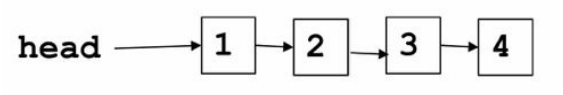
For examples: If N = 2, capacity = 8, it means the items array is too big and we have to shrink the array. The shrink method will create an array of size 4, and move the content of the items array to the new array.

```
public class Bag<E>{
      private int capacity;
      private int N;    //number of items
      private E[] items;

      // other methods are here

      private void shrink(){
                  //your code here
            if (N <= capacity / 4) {
                  E[] smaller = (E[])new Object[capacity / 2];
                  for (int i = 0; i < N; i++) {
                        smaller[i] = items[i];
                  }
                  items = smaller;
                  capacity /= 2;
            }
      }
}
```

2. [10 pts] Given a Node class

```
class Node{
      Integer data;
      Node next;
}
```

and a Node reference **`head`** references the first node in the list, write a method `“int sum(Node head)”` that calculates the sum of all data in a given linked list. Return 0 if the list is empty.
For example:



sum(head) will return 10

```
int sum(Node head){
      //your code here
      int total = 0;
      for (Node current = head; current != null; current = current.next) {
            total += current.data;
      }
      return total;
}
```

3. [10 pts] A Date class has three private members year, month, and day. We want to make the Date class iterable, so that we can iterate through the three values in year, month, and day order. Make the Data class Iterable by filling the blanks.

```
import java.util.Iterator;
import java.lang.*;
public class Date implements Iterable<Integer>{
  private int year,month,day;
  public Date(int y,int m,int d){
      year = y; month = m; day = d;
  }
  @Override
  public Iterator<Integer> iterator(){
      return new DateIterator()________________;
  }

  private class DateIterator implements Iterator<Integer>{
      int[] arr = {year, month, day}; int count = 0;
      @Override
      public boolean__________hasNext()_____________{

            return (count < 3);
      }

      @Override
      public Integer __________next()_______________{

            if (hasNext()) {

            _____return arr[count++];

            }__________________________

      }
  }
  public static void main(String[] s){
      Date date = new Date(2018,6,20);
      for(Integer d: date){
            System.out.print(d + ",");
      }
  }
}
// Output: 2018,6,20
```

4. [10 pts] Given the sorted array of integers, we want to find an item using the binary search algorithm. Write a method "`boolean binarySearch(Integer[ ] items, Integer item)`" that checks if the item exists in the items array using the binary search algorithm.

```
boolan binarySearch(Integer[] items, Integer item){


      int start = 0, end = items.length - 1, mid;
      while (start <= end) {
            mid = (start + end) / 2;
            if (items[mid] == item) {
                  return true;
            } else if (items[mid] > item) {
                  end = mid - 1;
            } else {
                  start = mid + 1;
            }
      }
      return false;


}
```