15.1 Which statements about the hashCode() and equals() methods are true?
Select the two correct answers.
(a) Two objects that are different according to the equals() method, must have different hash values.
(b) Two objects that are equal according to the equals() method, must have the same hash value.
(c) Two objects that have the same hash value, must be equal according to the equals() method.
(d) Two objects that have different hash values, must be unequal according to the equals() method.

15.2 Given that the objects referenced by the parameters override the equals() and the hashCode() methods appropriately, which return values are possible from the following method?

```
String func(Object x, Object y) {
 return (x == y) + " " + x.equals(y) + " " + (x.hashCode() == y.hashCode());
}
```

Select the four correct answers.
(a) "false false false"
(b) "false false true"
(c) "false true false"
(d) "false true true"
(e) "true false false"
(f) "true false true"
(g) "true true false"
(h) "true true true"

15.3 Which code, when inserted at (1), in the equalsImpl() method will provide a correct implementation of the equals() method?

```
public class Pair {
  int a, b;
  public Pair(int a, int b) {
    this.a = a;
    this.b = b;
  }
  public boolean equals(Object o) {
    return (this == o) || (o instanceof Pair) && equalsImpl((Pair) o);
  }
  private boolean equalsImpl(Pair o) {
    // (1) INSERT CODE HERE ...
  }
}
```

Select the three correct answers.

```
(a) return a == o.a || b == o.b;
(b) return false;
(c) return a >= o.a;
(d) return a == o.a;
(e) return a == o.a && b == o.b;
```

15.4 Which code, when inserted at (1), will provide a correct implementation of the hashCode() method in the following program?

```
import java.util.*;
public class Measurement {
  int count;
  int accumulated;
  public Measurement() {}
  public void record(int v) {
    count++;
    accumulated += v;
  }
  public int average() {
    return accumulated/count;
  }
  public boolean equals(Object other) {
    if (this == other)
      return true;
    if (!(other instanceof Measurement))
      return false;
    Measurement o = (Measurement) other;
    if (count != 0 && o.count != 0)
      return average() == o.average();
    return count == o.count;
  }
  public int hashCode() {
    // (1) INSERT CODE HERE ...
  }
}
```

Select the two correct answers.

```
(a) return 31337;
(b) return accumulated / count;
(c) return (count << 16) ^ accumulated;
(d) return ~accumulated;
(e) return count == 0 ? 0 : average();
```

15.5 What will be the result of compiling and running the following program?

```
import java.util.Comparator;
class Person implements Comparable<Person> {
  String name;
  int age;
  Person(String name, int age) { this.name = name; this.age = age; }
  public int compareTo(Person p2) {
    Comparator<String> strCmp = Person.cmp();
    int status = strCmp.compare(this.name, p2.name);
    if (status == 0) {
      Comparator<Integer> intCmp = Person.cmp();
      status = intCmp.compare(this.age, p2.age);
    }
    return status;
  }
  public static <E extends Comparable<E>> Comparator<E> cmp() {
    return new Comparator<E>() {
      public int compare(E s1, E s2) { return s2.compareTo(s1); }
    };
  }
  public static void main(String[] args) {
    Person p1 = new Person("Tom", 20);
    Person p2 = new Person("Dick", 30);
    Person p3 = new Person("Tom", 40);
    System.out.println((p1.compareTo(p2) < 0) + " " + (p1.compareTo(p3) < 0));
  }
}
```

Select the one correct answer.
(a) The program will fail to compile.
(b) The program will compile but throw an exception when run.
(c) The program will compile and print true false, when run.
(d) The program will compile and print true true, when run.
(e) The program will compile and print false false, when run.
(f) The program will compile and print false true, when run.

15.6 Which of these are core interfaces in the collections framework?
Select the three correct answers.
(a) Set<E>
(b) Bag<E>
(c) LinkedList<E>
(d) Collection<E>
(e) Map<K,V>

15.7 Which of these implementations are provided by the java.util package?
Select the two correct answers.
(a) HashList<E>
(b) HashMap<K,V>
(c) ArraySet<E>
(d) ArrayMap<K,V>
(e) TreeMap<K,V>

15.8 What is the name of the interface used to represent collections that maintain non-unique elements in order?
Select the one correct answer.
(a) Collection<E>
(b) Set<E>
(c) SortedSet<E>
(d) List<E>
(e) Sequence<E>

15.9 Which methods are specified by the Iterator<E> interface?
Select the three correct answers.
(a) hasNext()
(b) hasMore()
(c) remove()
(d) delete()
(e) more()
(f) next()

15.10 Which identifiers, when inserted in appropriate places in the program, will result in the output 911?

```
Collection<________> myItems = new ArrayList<__________>();
myItems.add(9); myItems.add(1); myItems.add(1);
Iterator<_________> iterator = _____________.iterator();
while (_____________.___________()) {
  System.out.print(_____________._________());
}
```

Select the five correct answers.
(a) hasNext
(b) myItems
(c) next
(d) Integer
(e) int
(f) Collection
(g) iterator

15.11 What will the program print when it is compiled and run?

```
import java.util.ArrayList;
import java.util.Collection;
public class RQ400_100 {
  public static void main(String[] args) {
    int sum = 0;
    for (int  i : makeCollection())
      sum += i;
    System.out.println(sum);
  }
  static Collection<Integer> makeCollection() {
    System.out.println("A collection coming up.");
    Collection<Integer> collection = new ArrayList<Integer>();
    collection.add(10); collection.add(20); collection.add(30);
    return collection;
  }
}
```

Select the one correct answer.
(a) A collection coming up.
60
(b) A collection coming up.
A collection coming up.
A collection coming up.
60
(c) The program does not compile.
(d) None of the above.

15.12 Which statements are true about the for(:) loop:

```
for ( type variable : expression ) statement
```

Select the three correct answers.
(a) The variable is only visible in the for(:) loop body.
(b) The expression is only evaluated once.
(c) The type of the expression must be java.lang.Iterable or an array type.
(d) Changing the value of the variable in the loop body affects the data structure represented by the expression.
(e) The loop runs backwards if the expression is negated as follows: ! expression.
(f) We can iterate over several data structures simultaneously in a for(:) loop.

15.13 What will the program print when compiled and run?

```
import java.util.ArrayList;
import java.util.Collection;
public class IterateOverCollection2 {
  public static void main(String[] args) {
    Collection<String> words = new ArrayList<String>();
    words.add("Don't"); words.add("change"); words.add("me!");
    System.out.println("Before: " + words);
    for (String word : words) {
      System.out.print(word.toUpperCase() + "_");
    }
    System.out.println();
    System.out.println("After:  " + words);
  }
}
```

Select the one correct answer.

(a) Before: [Don't, change, me!]
DON'T_CHANGE_ME!_
After: [DON'T, CHANGE, ME!]

(b) Before: [Don't, change, me!]
DON'T_CHANGE_ME!_
After: [Don't, change, me!]

(c) The program will throw a java.util.ConcurrentModificationException, when run.

(d) The program fails to compile.

15.14 Which code, when inserted at (1), will result in the following output:

Before: [Apple, Orange, Apple]
After: [Orange]
from the program when compiled and run?

```
import java.util.ArrayList;
import java.util.Iterator;
import java.util.List;
class Fruity {
  private String fName;
  Fruity(String fName) { this.fName = fName; }
  public void setName(String newName) { this.fName = newName; }
  public String toString() { return fName; }
  public boolean equals(Object other) {
    if (this == other) return true;
    if (!(other instanceof Fruity)) return false;
    return fName.equalsIgnoreCase(((Fruity)other).fName);
  }
}
public class RQ400_50 {
  public static void main(String[] args) {
    Fruity apple = new Fruity("Apple");
    Fruity orange = new Fruity("Orange");
    List<Fruity> list = new ArrayList<Fruity>();
    list.add(apple); list.add(orange); list.add(apple);
    System.out.println("Before: " + list);
    // (1) INSERT CODE HERE ...
    System.out.println("After:  " + list);
  }
}
```

Select the two correct answers.

```
(a) for (Fruity f : list) {
  if (f.equals(apple))
    list.remove(f);
}
(b) int i = 0;
for (Fruity f : list) {
  if (f.equals(apple))
    list.remove(i);
  i++;
}
(c) for (int j = 0; j < list.size(); j++) {
  Fruity f = list.get(j);
  if (f.equals(apple))
    list.remove(j);
}
(d) Iterator<Fruity> itr = list.iterator();
while (itr.hasNext()) {
  Fruity f = itr.next();
  if (f.equals(apple))
    itr.remove();
}
```

15.15 Which statement, when inserted independently at (1), will cause either a compile-time or a runtime error?

```
import java.util.ArrayList;
import java.util.Collection;
public class RQ400_200 {
  public static void main(String[] args) {
    Collection<Integer> intList = new ArrayList<Integer>();
    intList.add(2008); intList.add(2009); intList.add(2010);
    // (1) INSERT STATEMENT HERE!
  }
}
```

Select the four correct answers.

```
(a) Object[] objArray1 = intList.toArray();
(b) Integer[] intArray1 = intList.toArray();
(c) Number[] numArray1 = intList.toArray();
(d) Object[] objArray2 = intList.toArray(new Object[0]);
(e) Integer[] intArray2 = intList.toArray(new Integer[0]);
(f) Integer[] intArray3 = intList.toArray(new Number[0]);
(g) Number[] numArray2 = intList.toArray(new Number[0]);
(h) Number[] numArray3 = intList.toArray(new Integer[0]);
(i) Number[] numArray4 = intList.toArray(new Long[0]);
```

15.16 Which statements about collections are true?

Select the two correct answers.

(a) Some operations on a collection may throw an UnsupportedOperationException.

(b) Methods calling optional operations in a collection must either catch an UnsupportedOperationException or declare it in their throws clause.

(c) A List can have duplicate elements.

(d) An ArrayList can only accommodate a fixed number of elements.

(e) The Collection interface contains a method named get.

15.17 What will be the result of attempting to compile and run the following program?

```
import java.util.ArrayList;
import java.util.Collection;
import java.util.HashSet;
import java.util.LinkedList;
import java.util.TreeSet;
public class Sets {
  public static void main(String[] args) {
    HashSet<Integer> set1 = new HashSet<Integer>();
    addRange(set1, 1);
    ArrayList<Integer> list1 = new ArrayList<Integer>();
    addRange(list1, 2);
    TreeSet<Integer> set2 = new TreeSet<Integer>();
    addRange(set2, 3);
    LinkedList<Integer> list2 = new LinkedList<Integer>();
    addRange(list2, 5);
    set1.removeAll(list1);
    list1.addAll(set2);
    list2.addAll(list1);
    set1.removeAll(list2);
    System.out.println(set1);
  }
  static void addRange(Collection<Integer> col, int step) {
    for (int i = step*2; i<=25; i+=step)
      col.add(i);
  }
}
```

Select the one correct answer.
(a) The program will fail to compile, since operations are performed on incompatible collection implementations.
(b) The program will fail to compile, since the TreeSet referenced by set2 has not been given a Comparator to use when sorting its elements.
(c) The program will compile without error, but will throw an UnsupportedOperationException, when run.
(d) The program will compile without error and will print all primes below 25, when run.

15.18 Which of these methods are defined in the Collection<E> interface?
Select the three correct answers.
(a) add(E o)
(b) retainAll(Collection<?> c)
(c) get(int index)
(d) iterator()
(e) indexOf(Object o)

15.19 What will the following program print?
```
import java.util.ArrayList;
import java.util.List;
import java.util.ListIterator;
public class Iterate {
  public static void main(String[] args) {
    List<String> l = new ArrayList<String>();
    l.add("A"); l.add("B"); l.add("C"); l.add("D"); l.add("E");
    ListIterator<String> i = l.listIterator();
    i.next(); i.next(); i.next(); i.next();
    i.remove();
    i.previous(); i.previous();
    i.remove();
    System.out.println(l);
  }
}
```
Select the one correct answer.
(a) It will print [A, B, C, D, E].
(b) It will print [A, C, E].
(c) It will print [B, D, E].
(d) It will print [A, B, D].
(e) It will print [B, C, E].
(f) It will throw a NoSuchElementException.

15.20 Which sequence of digits will the following program print?
```
import java.util.ArrayList;
import java.util.LinkedList;
import java.util.List;
public class Lists {
  public static void main(String[] args) {
    List<String> list = new ArrayList<String>();
    list.add("1");
    list.add("2");
    list.add(1, "3");
    List<String> list2 = new LinkedList<String>(list);
    list.addAll(list2);
    list2 = list.subList(2, 5);
    list2.clear();
    System.out.println(list);
  }
}
```
Select the one correct answer.
(a) [1, 3, 2]
(b) [1, 3, 3, 2]
(c) [1, 3, 2, 1, 3, 2]
(d) [3, 1, 2]
(e) [3, 1, 1, 2]
(f) None of the above.

15.21 Which of these methods from the Collection interface will return the value true if the collection was modified during the operation?
Select the two correct answers.
(a) contains()
(b) add()
(c) containsAll()
(d) retainAll()
(e) clear()

15.22 Which statements, when inserted independently at (1), will guarantee that the following program will print [1, 9]?
```
import java.util.*;
public class RightOne {
  public static void main(String[] args) {
    // (1) INSERT DECLARATION HERE.
    collection.add(1); collection.add(9); collection.add(1);
    System.out.println(collection);
  }
}
```
Select the four correct answers.
(a) Collection<Integer> collection = new HashSet<Integer>();
(b) Set<Integer> collection = new HashSet<Integer>();
(c) HashSet<Integer> collection = new LinkedHashSet<Integer>();
(d) Set<Integer> collection = new LinkedHashSet<Integer>();
(e) Collection<Integer> collection = new TreeSet<Integer>();
(f) NavigableSet<Integer> collection = new TreeSet<Integer>();

15.23 What will the program print when it is compiled and run?
```
import static java.lang.System.out;
import java.util.Collections;
import java.util.NavigableSet;
import java.util.TreeSet;
public class RQ400_300 {
  public static void main(String[] args) {
    NavigableSet<String> strSetA = new TreeSet<String>();
    Collections.addAll(strSetA, "set", "shell", "soap");
    out.print(strSetA.ceiling("shell") + " ");
    out.print(strSetA.floor("shell") + " ");
    out.print(strSetA.higher("shell") + " ");
    out.println(strSetA.lower("shell"));
  }
}
```
Select the one correct answer.
(a) shell soap shell set
(b) soap set shell shell
(c) shell shell soap set
(d) set shell shell soap

15.24 Which statement, when inserted independently at (1), will result in program output that does not include the word shell?

```
import static java.lang.System.out;
import java.util.Collections;
import java.util.NavigableSet;
import java.util.TreeSet;
public class RQ400_400 {
  public static void main(String[] args) {
    NavigableSet<String> strSetA = new TreeSet<String>();
    Collections.addAll(strSetA, "set", "shell", "soap", "swan");
    // (1) INSERT STATEMENT HERE.
  }
}
```

Select the two correct answers.

(a) out.println(strSetA.headSet("soap", true));
(b) out.println(strSetA.headSet("soap", false));
(c) out.println(strSetA.tailSet("soap", true));
(d) out.println(strSetA.tailSet("soap", false));
(e) out.println(strSetA.subSet("set", false, "soap", true));
(f) out.println(strSetA.subSet("set", true, "soap", false));

15.25 Which collection types, when inserted at (1), will result in a generic method that will compile without errors?

```
public static <T> T justDoIt(_____/* (1) INSERT TYPE HERE */_____<T>
collection) {
  return collection.poll();
}
```

Select the three correct answers.

(a) NavigableSet
(b) PriorityQueue
(c) LinkedList
(d) Queue
(e) TreeSet
(f) LinkedHashSet

15.26 Which loop, when inserted independently at (1), will guarantee that the program will print sea|sells|she|shells|?

```
import static java.lang.System.out;
import java.util.Collections;
import java.util.PriorityQueue;
public class RQ400_500 {
  public static void main(String[] args) {
    PriorityQueue<String> strPQ = new PriorityQueue<String>();
    Collections.addAll(strPQ, "shells", "she", "sells", "sea");
    // (1) INSERT LOOP HERE
    out.println();
  }
}
```

Select the one correct answer.

```
(a) for (String word : strPQ) {
  out.print(word + "|");
}
(b) for (String word : strPQ) {
  out.print(strPQ.peek() + "|");
}
(c) while (strPQ.peek() != null) {
  out.print(strPQ.poll() + "|");
}
(d) for (String word : strPQ) {
 out.print(strPQ.poll() + "|");
}
```

15.27 Which of these methods can be called on objects implementing the Map<K, V> interface?

Select the two correct answers.

(a) contains(Object o)
(b) addAll(Map<? extends K, ? extends V> m)
(c) remove(Object o)
(d) values()
(e) toArray()

15.28 Which statements are true about maps?

Select the two correct answers.

(a) The return type of the values() method is Set.
(b) Changes made in the set view returned by keySet() will be reflected in the original map.
(c) The Map interface extends the Collection interface.
(d) All keys in a map are unique.
(e) All Map implementations keep the keys sorted.

15.29 Which of these classes have a comparator() method?

Select the two correct answers.

(a) ArrayList
(b) HashMap
(c) TreeSet
(d) HashSet
(e) TreeMap

15.30 Which methods are defined by the java.util.Map.Entry<K, V> interface?

Select the two correct answers.

(a) K getKey()
(b) K setKey(K value)
(c) V getValue()
(d) V setValue(V value)
(e) void set(K key, V value)

15.31 Which statements are true about the following program?

```
import java.util.*;
public class ConstructingSortedSets {
  public static void main(String[] args) {
    NavigableSet<Integer> navSet
       = new TreeSet<Integer>(Collections.reverseOrder());
    Collections.addAll(navSet, 2010, 3001, 2001);
    NavigableSet<Integer> ss1 = new TreeSet<Integer>(navSet);
    NavigableSet<Integer> ss2 = new
TreeSet<Integer>((Collection<Integer>)navSet);
    for (Integer iRef : navSet)                  // (1)
      System.out.print(iRef + "|");
    System.out.println();
    for (Integer iRef : ss1)                  // (2)
      System.out.print(iRef + "|");
    System.out.println();
    for (Integer iRef : ss2)                  // (3)
      System.out.print(iRef + "|");
    System.out.println();
    while (!ss1.isEmpty())                    // (4)
      System.out.print(ss1.pollFirst() + "|");
    System.out.println();
    while (!ss2.isEmpty())                    // (5)
      System.out.print(ss2.pollFirst() + "|");
  }
}
```

Select the three correct answers.

(a) The loop at (1) prints 3001|2010|2001|.

(b) The loops at (1), (2) and (4) print the same output.

(c) The loop at (3) prints 3001|2010|2001|.

(d) All the loops print the same output.

(e) The loops at (3) and (5) print the same output.

15.32 Which code, when inserted independently at (1), will result in the following output from the program: {be=2, not=1, or=1, to=2}?

```
import java.util.Map;
import java.util.TreeMap;
public class FreqMap {
  public static void main(String[] args) {
    Map<String, Integer> freqMap = new TreeMap<String, Integer>();
    for (String key : new String[] {"to", "be", "or", "not", "to", "be"}) {
      // (1) INSERT CODE HERE ...
    }
    System.out.println(freqMap);
  }
}
```

Select the two correct answers.

```
(a) Integer frequency = freqMap.get(key);
frequency = (frequency == 0) ? 1 : frequency+1;
freqMap.put(key, frequency);
(b) Integer frequency = freqMap.get(key);
frequency = (frequency == null) ? 1 : frequency+1;
freqMap.put(key, frequency);
(c) int frequency = freqMap.get(key);
frequency = (frequency == 0) ? 1 : frequency+1;
freqMap.put(key, frequency);
(d) Integer frequency = (!freqMap.containsKey(key)) ? 1 : freqMap.get(key)+1;
freqMap.put(key, frequency);
```

15.33 What will the program print when compiled and run?

```
import java.util.Collection;
import java.util.Map;
import java.util.NavigableMap;
import java.util.TreeMap;
public class MapModify {
  public static void main(String[] args) {
    NavigableMap<String, Integer> grades = new TreeMap<String, Integer>();
    grades.put("A",  5); grades.put("B", 10); grades.put("C", 15);
    grades.put("D", 20); grades.put("E", 25);
    System.out.printf("1:%d, ", grades.get(grades.firstKey()));
    System.out.printf("2:%d, ", sumValues(grades.headMap("D")));
    System.out.printf("3:%d,   ",   sumValues(grades.subMap("B",   false,   "D",
true)));
    grades.subMap(grades.firstKey(), false, grades.lastKey(), false).clear();
    System.out.printf("4:%d%n", sumValues(grades));
  }
  public static <K, M extends Map<K, Integer>> int sumValues(M freqMap) {
    Collection<Integer> values = freqMap.values();
    int sumValues= 0;
    for (int value : values)
      sumValues += value;
    return sumValues;
  }
}
```

Select the one correct answer.

(a) 1:5, 2:50, 3:35, 4:30

(b) 1:5, 2:30, 3:35, 4:30

(c) 1:5, 2:30, 3:25, 4:30

(d) 1:5, 2:30, 3:35, 4:75

15.34 Which code, when inserted independently at (1), will result in the following output from the program: {Soap=10, Salts=10}?

```
import java.util.*;
public class Mapping {
  public static void main(String[] args) {
    NavigableMap<String, Integer> myMap
 = new TreeMap<String, Integer>(Collections.reverseOrder());
    myMap.put("Soap", 10); myMap.put("Shampoo", 5); myMap.put("Salts", 10);
    // (1) INSERT CODE HERE ...
    System.out.println(myMap);
  }
}
```

Select the two correct answers.

```
(a) for (Map.Entry<String, Integer> entry : myMap.entrySet())
  if (entry.getKey().equals("Shampoo"))
    myMap.remove("Shampoo");
(b) for (Iterator<String> iterator = myMap.keySet().iterator();
     iterator.hasNext();)
  if (iterator.next().equals("Shampoo"))
    iterator.remove();
(c) for (Iterator<String> iterator = myMap.keySet().iterator();
     iterator.hasNext();) {
  if (iterator.next().equals("Shampoo"))
    myMap.remove("Shampoo");
(d) for (Map.Entry<String, Integer> entry : myMap.entrySet())
  if (entry.getKey().equals("Shampoo"))
    myMap.remove(entry);
(e) myMap.subMap("Shampoo", true, "Shampoo", true).clear();
```

15.35 Which code, when inserted independently at (1), will result in the following output from the program: {1=Odd, 2=Even, 3=Odd}?

```
import java.util.Map;
import java.util.TreeMap;
public class StringBuilderMap {
  public static void main(String[] args) {
    Map<Integer, StringBuilder> myMap = new TreeMap<Integer, StringBuilder>();
    for (Integer key : new int[] {1, 2, 1, 3, 1, 2, 3, 3}) {
      // (1) INSERT CODE HERE ...
    }
    System.out.println(myMap);
  }
  private static StringBuilder toggle(StringBuilder strBuilder) {
    String value = "Odd";
    if (strBuilder.toString().equals(value))
      value = "Even";
    return strBuilder.replace(0, strBuilder.length(), value);
  }
}
```

Select the one correct answer.

```
(a) StringBuilder value = myMap.get(key);
myMap.put(key, (value == null) ? new StringBuilder("Odd") :
     StringBuilderMap.toggle(value));
(b) StringBuilder value = myMap.get(key);
if (value == null)
  value = new StringBuilder("Odd");
else
  StringBuilderMap.toggle(value);
myMap.put(key, value);
(c) StringBuilder value = myMap.get(key);
if (!myMap.containsKey(key))
  myMap.put(key, new StringBuilder("Odd"));
else
  StringBuilderMap.toggle(value);
(d) All of the above.
```

15.36 Which code, when inserted independently at (1), will result in the following output from the program: {1=Odd, 2=Even, 3=Odd}?

```
import java.util.Map;
import java.util.TreeMap;
public class StringMap {
  public static void main(String[] args) {
    Map<Integer, String> myMap = new TreeMap<Integer, String>();
    for (Integer key : new int[] {1, 2, 1, 3, 1, 2, 3, 3}) {
      // (1) INSERT CODE HERE ...
    }
    System.out.println(myMap);
  }
  private static String toggle(String str) {
    if (str.equals("Odd"))
      str = str.replace("Odd", "Even");
    else
      str = str.replace("Even", "Odd");
    return str;
  }
}
```

Select the one correct answer.

```
(a) String value = myMap.get(key);
myMap.put(key, (value == null) ? "Odd" : StringMap.toggle(value));
(b) String value = myMap.get(key);
if (value == null)
  value = "Odd";
else
  StringMap.toggle(value);
myMap.put(key, value);
(c) String value = myMap.get(key);
if (!myMap.containsKey(key))
  myMap.put(key, "Odd");
else
  StringMap.toggle(value);
(d) All of the above.
```

15.37 Which statement is true about the following program?

```
import java.util.ArrayList;
import java.util.Collections;
import java.util.List;
public class WhatIsThis {
  public static void main(String[] args) {
    List<StringBuilder> list = new ArrayList<StringBuilder>();
    list.add(new StringBuilder("B"));
    list.add(new StringBuilder("A"));
    list.add(new StringBuilder("C"));
    Collections.sort(list, Collections.reverseOrder());
    System.out.println(list.subList(1,2));
  }
}
```

Select the one correct answer.

(a) The program will compile and print the following when run: [B].
(b) The program will compile and print the following when run: [B, A].
(c) The program will compile, but throw an exception when run.
(d) The program will not compile.

15.38 Which statement is true about the following program?

```
import java.util.ArrayList;
import java.util.Arrays;
import java.util.Collections;
import java.util.List;
public class WhatIsThat {
  public static void main(String[] args) {
    List<StringBuilder> list = new ArrayList<StringBuilder>();
    list.add(new StringBuilder("B"));
    list.add(new StringBuilder("A"));
    list.add(new StringBuilder("D"));
    list.add(new StringBuilder("C"));
    StringBuilder[] sbArray = list.toArray(new StringBuilder[0]);
    Collections.sort(list);
    Collections.sort(list, null);
    Collections.sort(list, Collections.reverseOrder());
    System.out.println("List: " + list);
    Arrays.sort(sbArray);
    Arrays.sort(sbArray, null);
    Arrays.sort(sbArray, Collections.reverseOrder());
    System.out.println("Array: " + Arrays.toString(sbArray));
  }
}
```

Select the one correct answer.
(a) The program will compile and print the following when run: [B].
(b) The program will compile and print the following when run: [B, A].
(c) The program will compile, but throw an exception when run.
(d) The program will not compile.

15.39 Which statements are true about the following program?

```
import java.util.Arrays;
public class GetThatIndex {
  public static void main(String[] args) {
    if (args.length != 1) return;
    printIndex(args[0]);
  }
  public static void printIndex(String key) {
    String[] strings = {"small", "smaller", "smallest", "tiny"};
    System.out.println(Arrays.binarySearch(strings , key));
  }
}
```

Select the two correct answers.
(a) The largest value ever printed by the printIndex() method is 3.
(b) The largest value ever printed by the printIndex() method is 4.
(c) The largest value ever printed by the printIndex() method is 5.
(d) The smallest value ever printed by the printIndex() method is 0.
(e) The smallest value ever printed by the printIndex() method is -4.
(f) The smallest value ever printed by the printIndex() method is -5.
(g) The smallest value ever printed by the printIndex() method is -3.

15.40 Given that the following program compiles and prints the following output when run: 1 1 1.

```
import java.util.*;
public class SoulSearching {
  public static void main(String[] args) {
    String[] array = {"smallest", "small", "tiny", "smaller"};
    List<String> list1 = Arrays.___________(array);
    Collections.__________(list1);
    int index1 = Collections.______________(list1, "smaller");
    System.out.print(index1 + " ");
    List<String> list2 = Arrays.___________(array);
    Collections.__________(list2);
    int index2 = list2.__________("smaller");
    System.out.print(index2 + " ");
    Arrays._________(array);
    int index3 = Arrays.____________(array, "smaller");
    System.out.println(index3);
  }
}
```

Which method names can be used to fill the blanks without violating the behavior of the program?
Select the four correct answers.

(a) asList	(b) contains
(c) sort	(d) findIndex
(e) indexOf	(f) binarySearch
(g) search	(h) toList
(i) toArray	(j) subList

15.41 What will the program print when compiled and run?

```
import java.util.Arrays;
import java.util.Collections;
import java.util.Comparator;
import java.util.List;
public class LastOrders {
  public static void main(String[] args) {
    String[] array = {"slurs", "slush", "slurps", "slurry"};
    List<String> list1 = Arrays.asList(array);
    Collections.sort(list1, LastOrders.comparatorX());
    int index1 = Collections.binarySearch(list1, "slurry",
LastOrders.comparatorX());
    System.out.println (list1 + ": slurry at " + index1);
  }
  public static Comparator<String> comparatorX() {
    return new Comparator<String>() {
      public int compare(String str1, String str2) {
        StringBuilder sb1 = new StringBuilder(str1);
        StringBuilder sb2 = new StringBuilder(str2);
        return sb2.reverse().toString().compareTo(sb1.reverse().toString());
      }
    };
  }
}
```

Select the one correct answer.
(a) [slush, slurs, slurry, slurps]: slurry at 2
(b) [slush, slurps, slurs, slurry]: slurry at 3
(c) [slurry, slurs, slurps, slush]: slurry at 0
(d) [slurps, slurry, slurs, slush]: slurry at 1