Mock Test#2

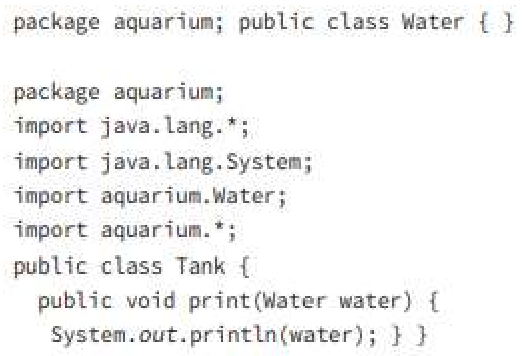
Total points 37/50

The respondent's email ( ) was recorded on submission of this form.

Last Name

First Name

1. Given the following classes, what is the maximum number of imports that can be removed and have the code still compile? *0/1



```
package aquarium; public class Water { }

package aquarium;
import java.lang.*;
import java.lang.System;
import aquarium.Water;
import aquarium.*;
public class Tank {
  public void print(Water water) {
   System.out.println(water); } }
```

☐ A. 0

☐ B. 1

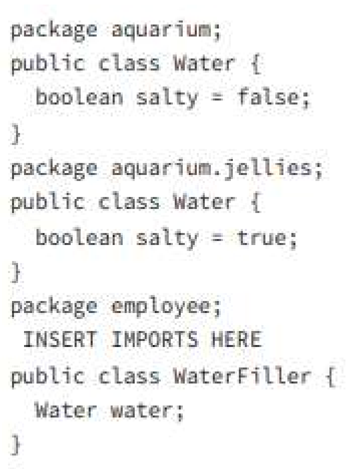
☐ C. 2

☐ D. 3

☑ E. 4

☐ F. Does not compile.

2.Given the following classes, which of the following snippets can be inserted in place of INSERT IMPORTS HERE and have the code compile? (Choose all that apply) *0/1



```
package aquarium;
public class Water {
  boolean salty = false;
}
package aquarium.jellies;
public class Water {
  boolean salty = true;
}
package employee;
 INSERT IMPORTS HERE
public class WaterFiller {
  Water water;
}
```

☐ A. import aquarium.*;

☐ B. import aquarium.Water; import aquarium.jellies.*;

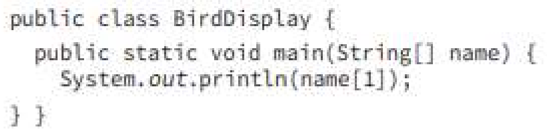
☑ C. import aquarium.*; import aquarium.jellies.Water;

☑ D. import aquarium.*; import aquarium.jellies.*;

☐ E. import aquarium.Water; import aquarium.jellies.Water;

☐ F. None of these imports can make the code compile.

3.Given the following class, which of the following calls print out Blue Jay? (Choose all that apply) *1/1



```
public class BirdDisplay {
  public static void main(String[] name) {
    System.out.println(name[1]);
} }
```

☐ A. java BirdDisplay Sparrow Blue Jay

☑ B. java BirdDisplay Sparrow "Blue Jay"

☐ C. java BirdDisplay Blue Jay Sparrow

☐ D. java BirdDisplay "Blue Jay" Sparrow

☐ E. java BirdDisplay.class Sparrow "Blue Jay"

☐ F. java BirdDisplay.class "Blue Jay" Sparrow

☐ G. Does not compile.

4.Which of the following legally fill in the blank so you can run the main() method from the command line? (Choose all that apply) *0/1

public static void main( )

☑ A. String[] _names

☐ B. String[] 123

☑ C. String abc[]

☑ D. String _Names[]

☑ E. String... $n

☐ F. String names

☐ G. None of the above.

5.Which of the following are legal entry point methods that can be run from the command line? (Choose all that apply) * 1/1

□ A. private static void main(String[] args)

□ B. public static final main(String[] args)

□ C. public void main(String[] args)

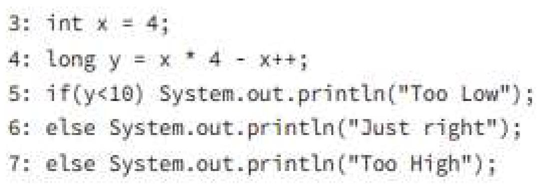
□ D. public static void test(String[] args)

☑ E. public static void main(String[] args)

□ F. public static main(String[] args)

□ G. None of the above

6.What is the output of the following code snippet? * 1/1



```
3: int x = 4;
4: long y = x * 4 - x++;
5: if(y<10) System.out.println("Too Low");
6: else System.out.println("Just right");
7: else System.out.println("Too High");
```

□ A. Too Low

□ B. Just Right

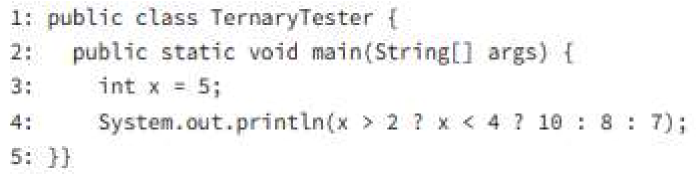
□ C. Too High

□ D. Compiles but throws a NullPointerException.

□ E. The code will not compile because of line 6.

☑ F. The code will not compile because of line 7.

7.What is the output of the following code? * 1/1



```
1: public class TernaryTester {
2:   public static void main(String[] args) {
3:     int x = 5;
4:     System.out.println(x > 2 ? x < 4 ? 10 : 8 : 7);
5: }}
```

☐ A. 5

☐ B. 4

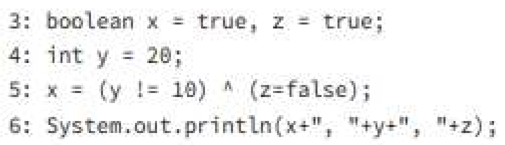
☐ C. 10

☑ D. 8

☐ E. 7

☐ F. The code will not compile because of line 4

8.What is the output of the following code snippet? * 1/1



```
3: boolean x = true, z = true;
4: int y = 20;
5: x = (y != 10) ^ (z=false);
6: System.out.println(x+", "+y+", "+z);
```

☐ A. true, 10, true

☑ B. true, 20, false

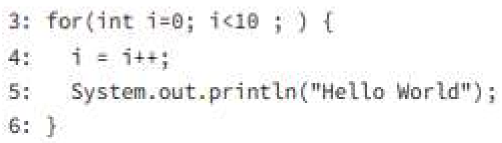
☐ C. false, 20, true

☐ D. false, 20, false

☐ E. false, 20, true

☐ F. The code will not compile because of line 5.

9.How many times will the following code print "Hello World"? * 1/1



```
3: for(int i=0; i<10 ; ) {
4:    i = i++;
5:    System.out.println("Hello World");
6: }
```

□ A. 9

□ B. 10

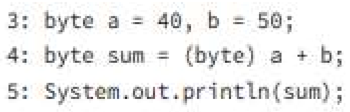
□ C. 11

□ D. The code will not compile because of line 3.

□ E. The code will not compile because of line 5.

☑ F. The code contains an infinite loop and does not terminate

10.What is the output of the following code? * 1/1



```
3: byte a = 40, b = 50;
4: byte sum = (byte) a + b;
5: System.out.println(sum);
```

□ A. 40

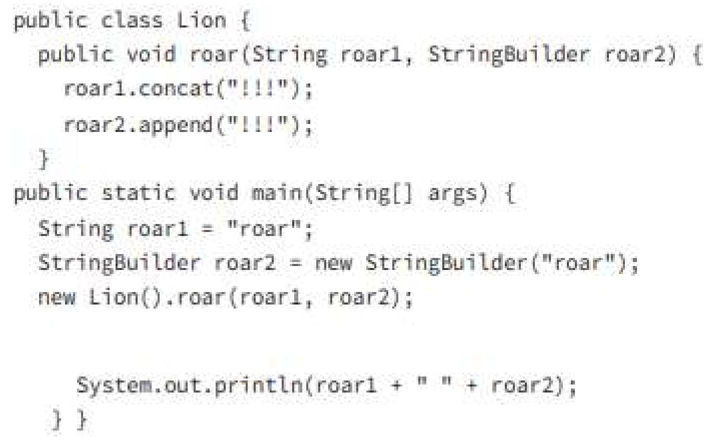
□ B. 50

□ C. 90

☑ D. The code will not compile because of line 4.

□ E. An undefined value.

11.What is the result of the following code? * 1/1



```
public class Lion {
  public void roar(String roar1, StringBuilder roar2) {
    roar1.concat("!!!");
    roar2.append("!!!");
  }
public static void main(String[] args) {
  String roar1 = "roar";
  StringBuilder roar2 = new StringBuilder("roar");
  new Lion().roar(roar1, roar2);


     System.out.println(roar1 + " " + roar2);
   } }
```

☐ A. roar roar

☑ B. roar roar!!!

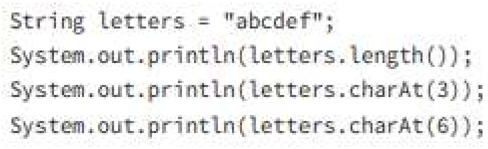
☐ C. roar!!! roar

☐ D. roar!!! roar!!!

☐ E. An exception is thrown.

☐ F. The code does not compile

12.Which are the results of the following code? (Choose all that apply) * 0/1



```
String letters = "abcdef";
System.out.println(letters.length());
System.out.println(letters.charAt(3));
System.out.println(letters.charAt(6));
```

☐ A. 5

☑ B. 6

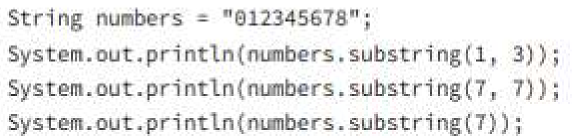
☐ C. c

☑ D. d

☑ E. An exception is thrown.

☐ F. The code does not compile

13.Which are the results of the following code? (Choose all that apply) * 1/1



```
String numbers = "012345678";
System.out.println(numbers.substring(1, 3));
System.out.println(numbers.substring(7, 7));
System.out.println(numbers.substring(7));
```

☑ A. 12

□ B. 123

□ C. 7

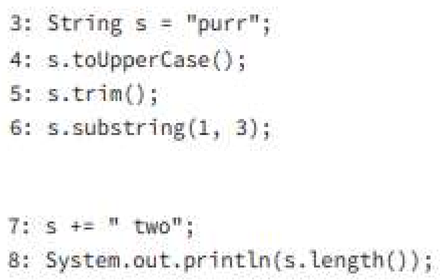
☑ D. 78

☑ E. A blank line.

□ F. An exception is thrown.

□ G. The code does not compile

14.What is the result of the following code? 3 * 0/1



```
String s = "purr";
s.toUpperCase();
s.trim();
s.substring(1, 3);

s += " two";
System.out.println(s.length());
```

☐ A. 2

☐ B. 4

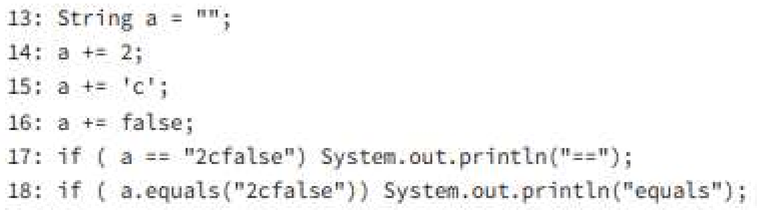
☐ C. 8

☐ D. 10

☐ E. An exception is thrown.

☑ F. The code does not compile.

15.What is the result of the following code? (Choose all that apply) * 0/1



```
13: String a = "";
14: a += 2;
15: a += 'c';
16: a += false;
17: if ( a == "2cfalse") System.out.println("==");
18: if ( a.equals("2cfalse")) System.out.println("equals");
```

☐ A. Compile error on line 14.

☑ B. Compile error on line 15.

☑ C. Compile error on line 16.

☐ D. Compile error on another line.

☐ E. ==

☐ F. equals

☐ G. An exception is thrown.

16.Which of the following are true? (Choose all that apply) * 0/1

☑ A. Package private access is more lenient than protected access.

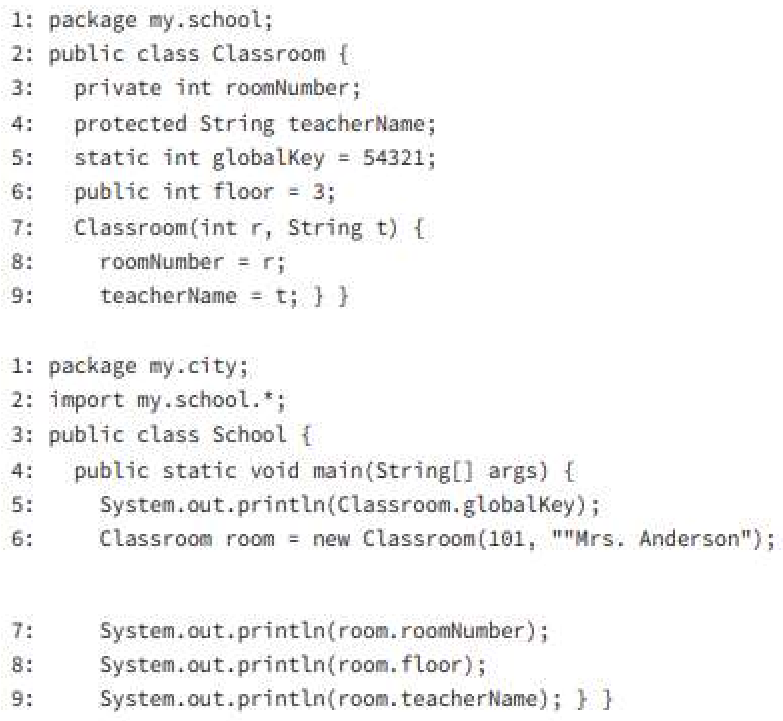
□ B. A public class that has private fields and package private methods is not visible to classes outside the package.

□ C. You can use access modifiers so only some of the classes in a package see a particular package private class.

□ D. You can use access modifiers to allow read access to all methods, but not any instance variables.

□ E. You can use access modifiers to restrict read access to all classes that begin with the word Test.

17.Given the following my.school.ClassRoom and my.city.School class definitions, which line numbers in main() generate a compiler error? (Choose all that apply) *0/1



```
1: package my.school;
2: public class Classroom {
3:   private int roomNumber;
4:   protected String teacherName;
5:   static int globalKey = 54321;
6:   public int floor = 3;
7:   Classroom(int r, String t) {
8:     roomNumber = r;
9:     teacherName = t; } }

1: package my.city;
2: import my.school.*;
3: public class School {
4:   public static void main(String[] args) {
5:     System.out.println(Classroom.globalKey);
6:     Classroom room = new Classroom(101, ""Mrs. Anderson");


7:     System.out.println(room.roomNumber);
8:     System.out.println(room.floor);
9:     System.out.println(room.teacherName); } }
```

□ A. None, the code compiles fine.

☑ B. Line 5

□ C. Line 6

□ D. Line 7

□ E. Line 8

□ F. Line 9

18.Which of the following are true? (Choose all that apply) * 1/1

☐ A. Encapsulation uses package private instance variables.

☑ B. Encapsulation uses private instance variables.

☑ C. Encapsulation allows setters.

☐ D. Immutability uses package private instance variables.

☑ E. Immutability uses private instance variables.

☐ F. Immutability allows setters.

19.Which are methods using JavaBeans naming conventions for accessors and mutators? (Choose all that apply) *1/1

☐ A. public boolean getCanSwim() { return canSwim;}

☐ B. public boolean canSwim() { return numberWings;}

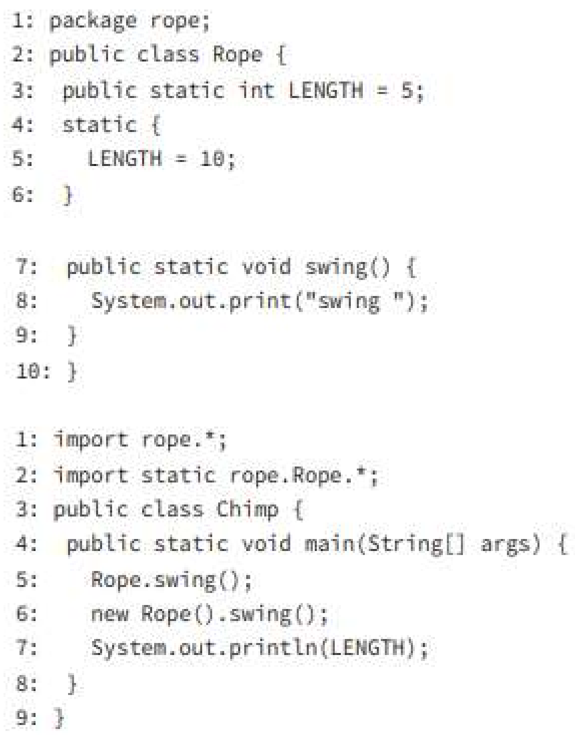
☑ C. public int getNumWings() { return numberWings;}

☐ D. public int numWings() { return numberWings;}

☑ E. public void setCanSwim(boolean b) { canSwim = b;}

20.What is the output of the following code? *

1/1



```
1: package rope;
2: public class Rope {
3:  public static int LENGTH = 5;
4:  static {
5:    LENGTH = 10;
6:  }

7:  public static void swing() {
8:    System.out.print("swing ");
9:  }
10: }

1: import rope.*;
2: import static rope.Rope.*;
3: public class Chimp {
4:  public static void main(String[] args) {
5:    Rope.swing();
6:    new Rope().swing();
7:    System.out.println(LENGTH);
8:  }
9: }
```

☐ A. swing swing 5

☑ B. swing swing 10

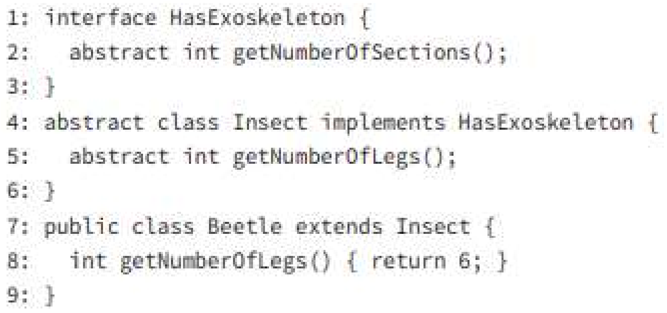
☐ C. Compiler error on line 2 of Chimp.

☐ D. Compiler error on line 5 of Chimp.

☐ E. Compiler error on line 6 of Chimp.

☐ F. Compiler error on line 7 of Chimp.

21.Choose the correct statement about the following code: * 1/1



```
1: interface HasExoskeleton {
2:   abstract int getNumberOfSections();
3: }
4: abstract class Insect implements HasExoskeleton {
5:   abstract int getNumberOfLegs();
6: }
7: public class Beetle extends Insect {
8:   int getNumberOfLegs() { return 6; }
9: }
```

☐ A. It compiles and runs without issue.

☐ B. The code will not compile because of line 2.

☐ C. The code will not compile because of line 4.

☑ D. The code will not compile because of line 7.

☐ E. It compiles but throws an exception at runtime.

22.Which of the following statements about polymorphism are true? (Choose all that apply) *1/1

☐ A. A reference to an object may be cast to a subclass of the object without an explicit cast.

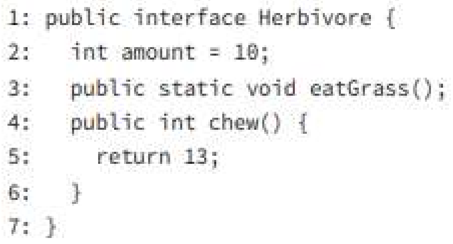
☑ B. If a method takes a superclass of three objects, then any of those classes may be passed as a parameter to the method.

☑ C. A method that takes a parameter with type java.lang.Object will take any reference.

☐ D. All cast exceptions can be detected at compile-time.

☐ E. By defining a public instance method in the superclass, you guarantee that the specific method will be called in the parent class at runtime.

23.Choose the correct statement about the following code: * 1/1



```
1: public interface Herbivore {
2:   int amount = 10;
3:   public static void eatGrass();
4:   public int chew() {
5:     return 13;
6:   }
7: }
```

☐ A. It compiles and runs without issue.

☐ B. The code will not compile because of line 2.

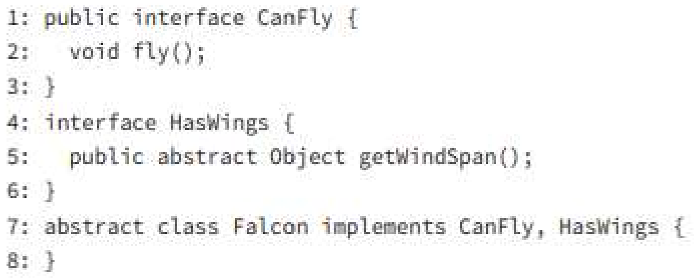
☐ C. The code will not compile because of line 3.

☐ D. The code will not compile because of line 4.

☐ E. The code will not compile because of lines 2 and 3.

☑ F. The code will not compile because of lines 3 and 4

24.Choose the correct statement about the following code: * 1/1



```
1: public interface CanFly {
2:   void fly();
3: }
4: interface HasWings {
5:   public abstract Object getWindSpan();
6: }
7: abstract class Falcon implements CanFly, HasWings {
8: }
```

☑ A. It compiles without issue.

☐ B. The code will not compile because of line 2.

☐ C. The code will not compile because of line 4.

☐ D. The code will not compile because of line 5.

☐ E. The code will not compile because of lines 2 and 5.

☐ F. The code will not compile because the class Falcon doesn't implement the interface methods

25.Which statements are true for both abstract classes and interfaces? (Choose all that apply) *1/1

□ A. All methods within them are assumed to be abstract.

☑ B. Both can contain public static final variables.

☑ C. Both can be extended using the extend keyword.

□ D. Both can contain default methods.

☑ E. Both can contain static methods.

☑ F. Neither can be instantiated directly.

□ G. Both inherit java.lang.Object.

26. What will happen if you add the statement System.out.println(5 / 0); to a working main() method? *1/1

□ A. It will not compile.

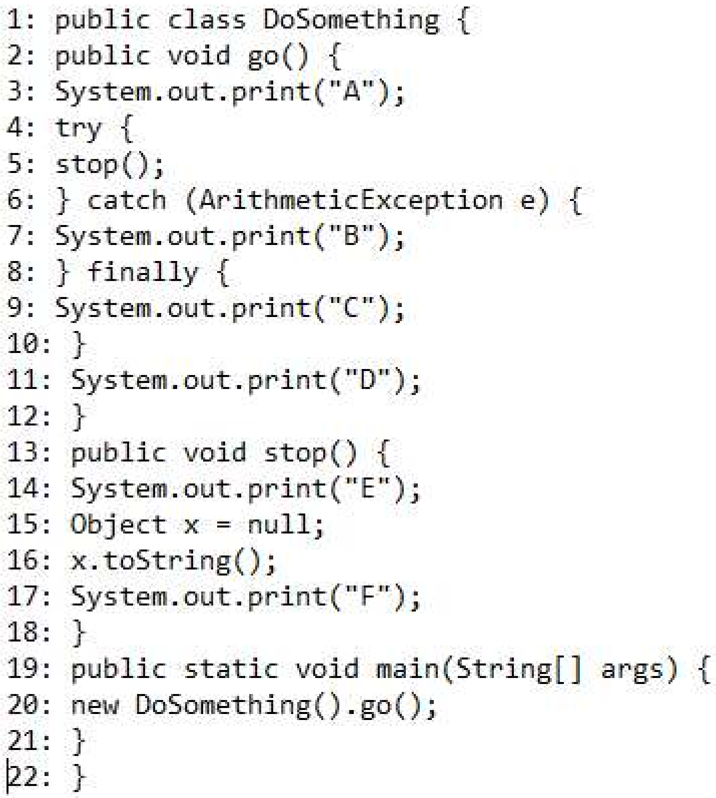
□ B. It will not run.

☑ C. It will run and throw an ArithmeticException.

□ D. It will run and throw an IllegalArgumentException.

□ E. None of the above.

27.What is printed besides the stack trace caused by the NullPointerException from line 16? *1/1



```
1: public class DoSomething {
2: public void go() {
3: System.out.print("A");
4: try {
5: stop();
6: } catch (ArithmeticException e) {
7: System.out.print("B");
8: } finally {
9: System.out.print("C");
10: }
11: System.out.print("D");
12: }
13: public void stop() {
14: System.out.print("E");
15: Object x = null;
16: x.toString();
17: System.out.print("F");
18: }
19: public static void main(String[] args) {
20: new DoSomething().go();
21: }
22: }
```

□ A. AE

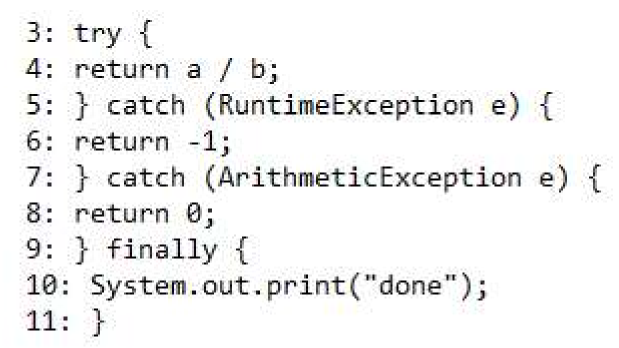
□ B. AEBCD

☑ C. AEC

□ D. AECD

□ E. No output appears other than the stack trace.

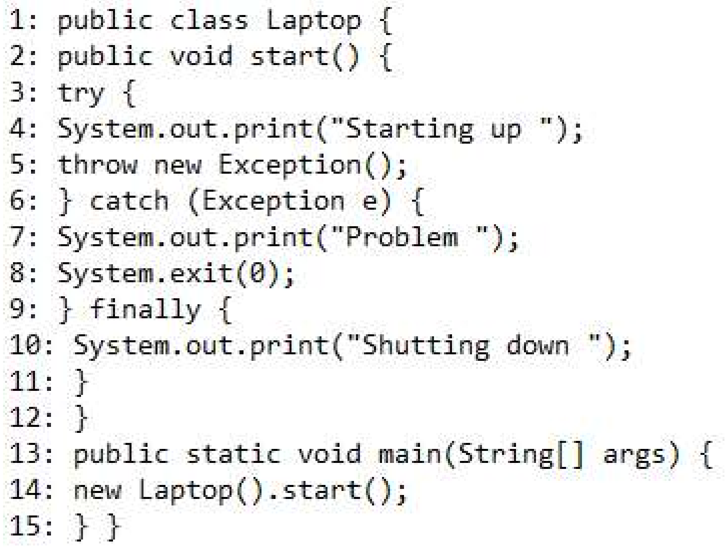
28.What is the output of the following snippet, assuming a and b are both 0? * 1/1



```
try {
return a / b;
} catch (RuntimeException e) {
return -1;
} catch (ArithmeticException e) {
return 0;
} finally {
System.out.print("done");
}
```

- □ A. -1
- □ B. 0
- □ C. done-1
- □ D. done0
- ☑ E. The code does not compile.
- □ F. An uncaught exception is thrown.

29.What is the output of the following program? * 1/1



```
public class Laptop {
public void start() {
try {
System.out.print("Starting up ");
throw new Exception();
} catch (Exception e) {
System.out.print("Problem ");
System.exit(0);
} finally {
System.out.print("Shutting down ");
}
}
public static void main(String[] args) {
new Laptop().start();
} }
```

□ A. Starting up

☑ B. Starting up Problem

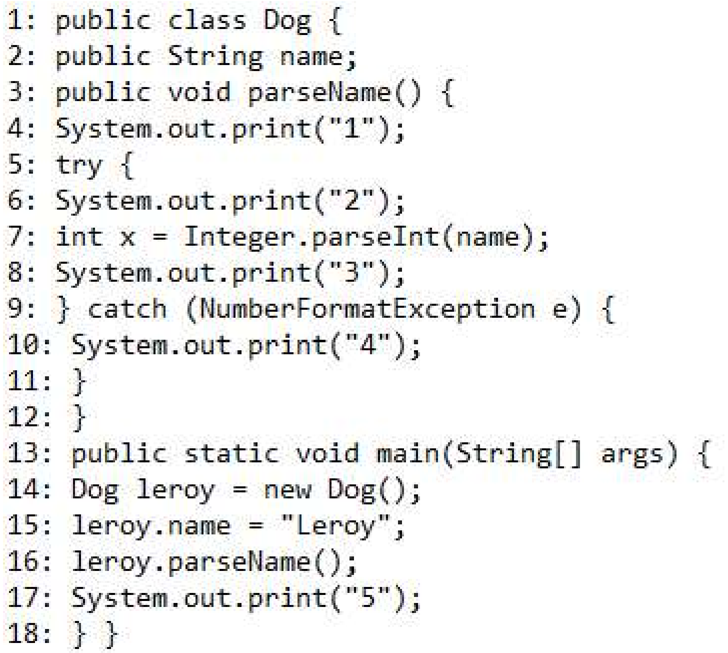
□ C. Starting up Problem Shutting down

□ D. Starting up Shutting down

□ E. The code does not compile.

□ F. An uncaught exception is thrown.

30.What is the output of the following program? * 0/1



```
public class Dog {
public String name;
public void parseName() {
System.out.print("1");
try {
System.out.print("2");
int x = Integer.parseInt(name);
System.out.print("3");
} catch (NumberFormatException e) {
System.out.print("4");
}
}
public static void main(String[] args) {
Dog leroy = new Dog();
leroy.name = "Leroy";
leroy.parseName();
System.out.print("5");
} }
```

□ A. 12

□ B. 1234

□ C. 1235

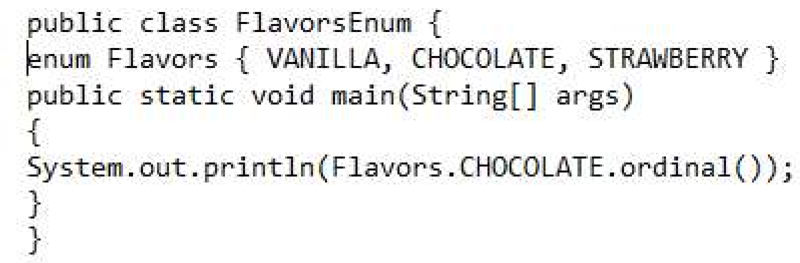
□ D. 124 E. 1245

□ E. 1245

□ F. The code does not compile.

☑ G. An uncaught exception is thrown.

31.What is the result of the following code? * 1/1



```
public class FlavorsEnum {
enum Flavors { VANILLA, CHOCOLATE, STRAWBERRY }
public static void main(String[] args)
{
System.out.println(Flavors.CHOCOLATE.ordinal());
}
}
```

□ A. 0

☑ B. 1

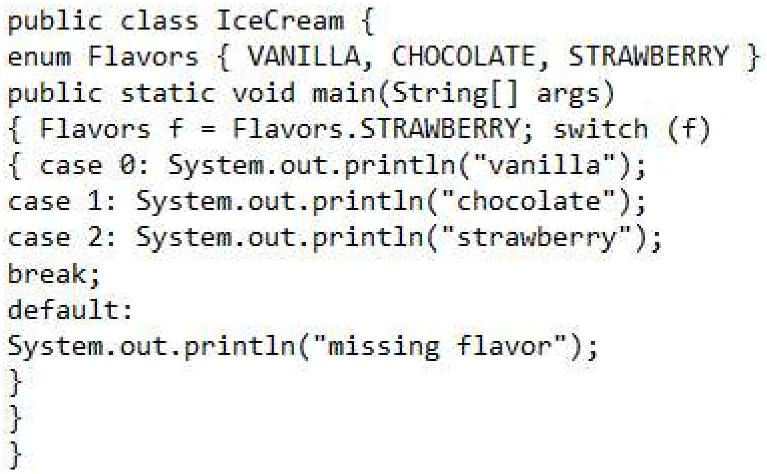
□ C. 9

□ D. CHOCOLATE

□ E. The code does not compile due to a missing semicolon.

□ F. The code does not compile for a different reason.

32.What is the result of the following code? (Choose all that apply.) * 1/1



```
public class IceCream {
enum Flavors { VANILLA, CHOCOLATE, STRAWBERRY }
public static void main(String[] args)
{ Flavors f = Flavors.STRAWBERRY; switch (f)
{ case 0: System.out.println("vanilla");
case 1: System.out.println("chocolate");
case 2: System.out.println("strawberry");
break;
default:
System.out.println("missing flavor");
}
}
}
```

□ A. vanilla

□ B. chocolate

□ C. strawberry

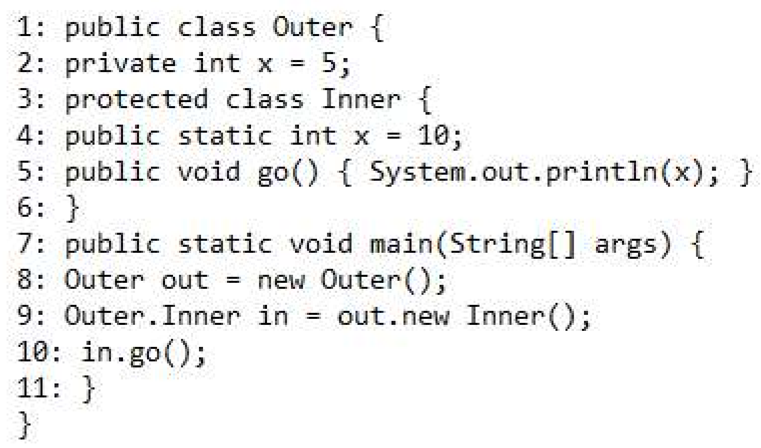
□ D. missing flavor

☑ E. The code does not compile.

□ F. An exception is thrown.

33.What is the result of the following code?

1/1



```
1: public class Outer {
2: private int x = 5;
3: protected class Inner {
4: public static int x = 10;
5: public void go() { System.out.println(x); }
6: }
7: public static void main(String[] args) {
8: Outer out = new Outer();
9: Outer.Inner in = out.new Inner();
10: in.go();
11: }
}
```

☐ A. The output is 5.

☐ B. The output is 10.

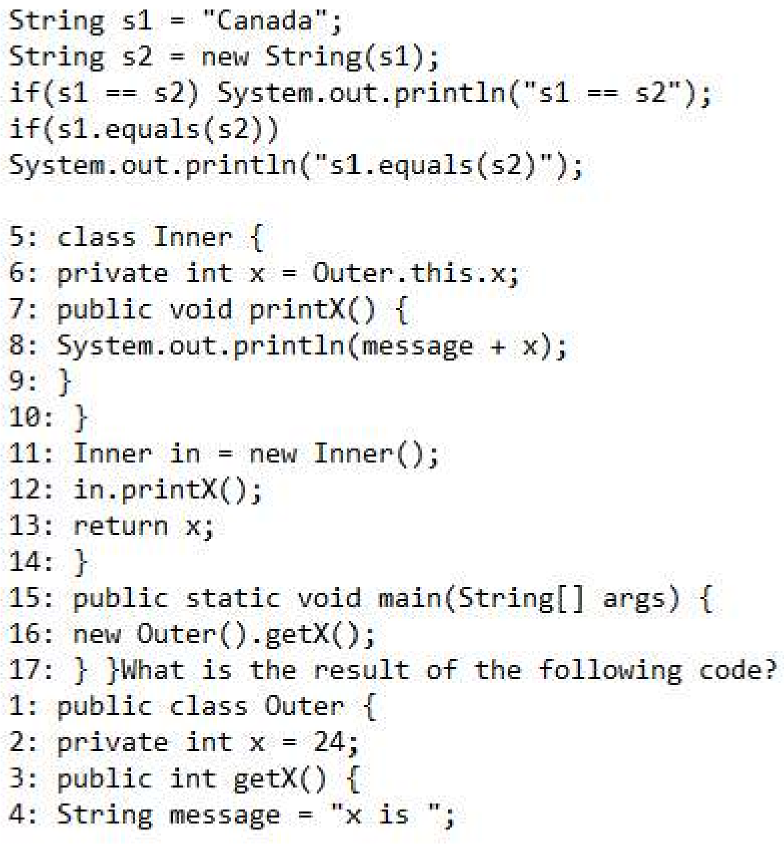
☑ C. Line 4 generates a compiler error.

☐ D. Line 8 generates a compiler error.

☐ E. Line 9 generates a compiler error.

☐ F. An exception is thrown.

34.What is the result of the following code? * 1/1



```
String s1 = "Canada";
String s2 = new String(s1);
if(s1 == s2) System.out.println("s1 == s2");
if(s1.equals(s2))
System.out.println("s1.equals(s2)");

5: class Inner {
6: private int x = Outer.this.x;
7: public void printX() {
8: System.out.println(message + x);
9: }
10: }
11: Inner in = new Inner();
12: in.printX();
13: return x;
14: }
15: public static void main(String[] args) {
16: new Outer().getX();
17: } }What is the result of the following code?
1: public class Outer {
2: private int x = 24;
3: public int getX() {
4: String message = "x is ";
```

□ A. x is 0.

□ B. x is 24.

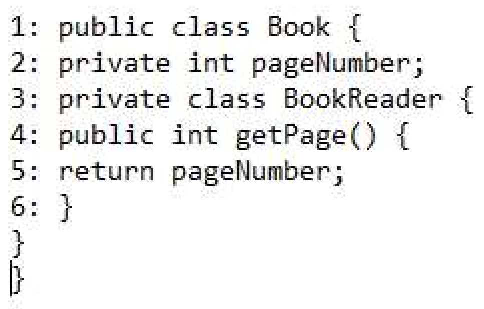
□ C. Line 6 generates a compiler error.

☑ D. Line 8 generates a compiler error.

□ E. Line 11 generates a compiler error.

□ F. An exception is thrown

35.The following code appears in a file named Book.java. What is the result of compiling the source file? *1/1



```
1: public class Book {
2: private int pageNumber;
3: private class BookReader {
4: public int getPage() {
5: return pageNumber;
6: }
}
}
```

☐ A. The code compiles successfully, and one bytecode file is generated: Book.class.

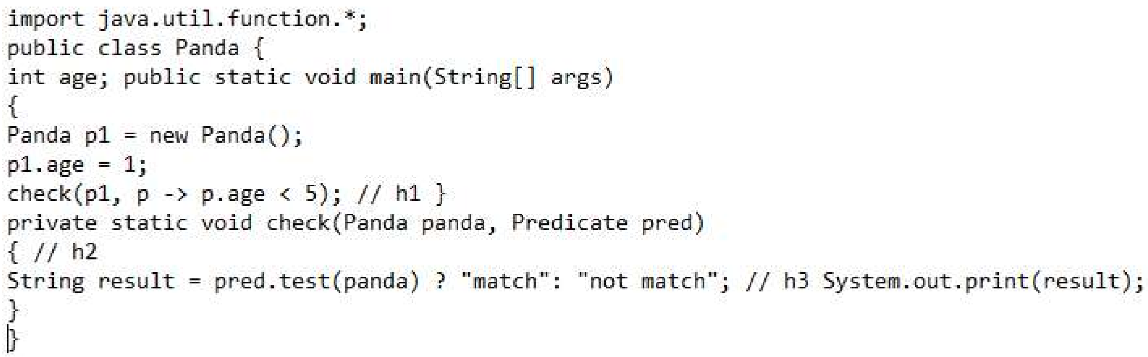
☐ B. The code compiles successfully, and two bytecode files are generated: Book.class and BookReader.class.

☑ C. The code compiles successfully, and two bytecode files are generated: Book.class and Book$BookReader.class.

☐ D. A compiler error occurs on line 3.

☐ E. A compiler error occurs on line 5.

36.What is the result of the following class? * 1/1



```
import java.util.function.*;
public class Panda {
int age; public static void main(String[] args)
{
Panda p1 = new Panda();
p1.age = 1;
check(p1, p -> p.age < 5); // h1 }
private static void check(Panda panda, Predicate pred)
{ // h2
String result = pred.test(panda) ? "match": "not match"; // h3 System.out.print(result);
}
}
```

☑ A. match

☐ B. not match

☐ C. Compiler error on line h1.

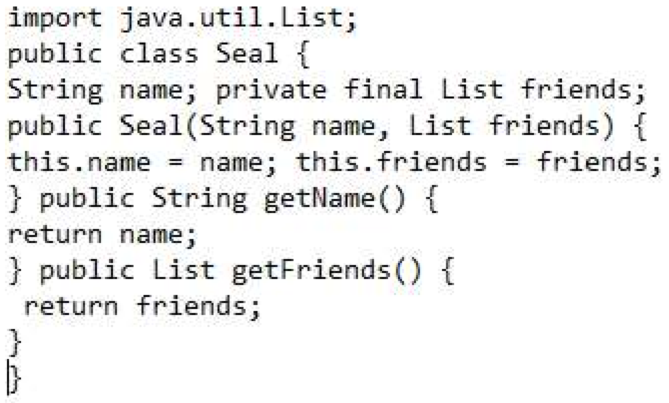
☐ D. Compiler error on line h2.

☐ E. Compile error on line h3.

☐ F. A runtime exception is thrown.

37.What changes need to be made to make the following immutable object pattern correct? (Choose all that apply.)

0/1



```
import java.util.List;
public class Seal {
String name; private final List friends;
public Seal(String name, List friends) {
this.name = name; this.friends = friends;
} public String getName() {
return name;
} public List getFriends() {
 return friends;
}
}
```

☐ A. None; the immutable object pattern is properly implemented.

☐ B. Have Seal implement the Immutable interface.

☑ C. Mark name final and private.

☐ D. Add setters for name and List friends.

☑ E. Replace the getFriends() method with methods that do not give the caller direct access to the List friends.

☐ F. Change the type of List to List.

☑ G. Make a copy of the List friends in the constructor.

☑ H. Mark the Seal class final.

38.Which of the following are true of interfaces? (Choose all that apply.) * 1/1

□ A. They can extend other classes.

□ B. They cannot be extended.

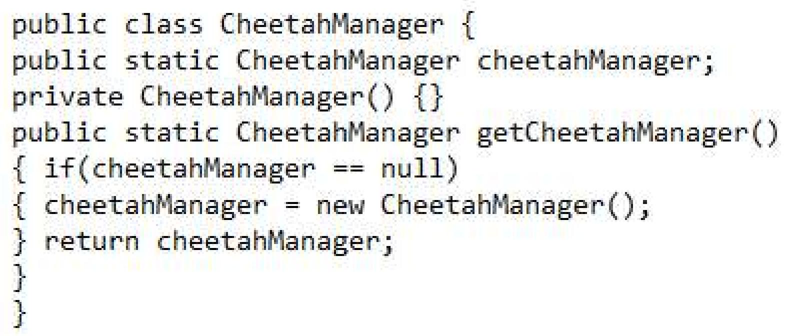
☑ C. They enable classes to have multiple inheritance.

□ D. They can only contain abstract methods.

□ E. They can be declared final.

☑ F. All members of an interface are public.

39.What changes need to be made to make the following singleton pattern correct? (Choose all that apply.) *0/1



```
public class CheetahManager {
public static CheetahManager cheetahManager;
private CheetahManager() {}
public static CheetahManager getCheetahManager()
{ if(cheetahManager == null)
{ cheetahManager = new CheetahManager();
} return cheetahManager;
}
}
```

□ A. None; the singleton pattern is properly implemented.

□ B. Rename cheetahManager to instance.

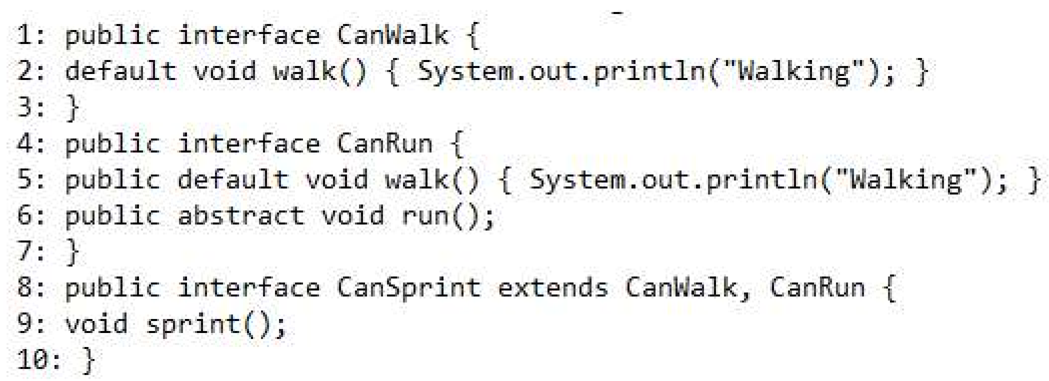
□ C. Rename getCheetahManager() to getInstance().

☑ D. Change the access modifier of cheetahManager from public to private.

□ E. Mark cheetahManager final.

☑ F. Add synchronized to getCheetahManager().

40.What is the result of the following code? * 0/1



```
1: public interface CanWalk {
2: default void walk() { System.out.println("Walking"); }
3: }
4: public interface CanRun {
5: public default void walk() { System.out.println("Walking"); }
6: public abstract void run();
7: }
8: public interface CanSprint extends CanWalk, CanRun {
9: void sprint();
10: }
```

☐ A. The code compiles without issue.

☐ B. The code will not compile because of line 5.

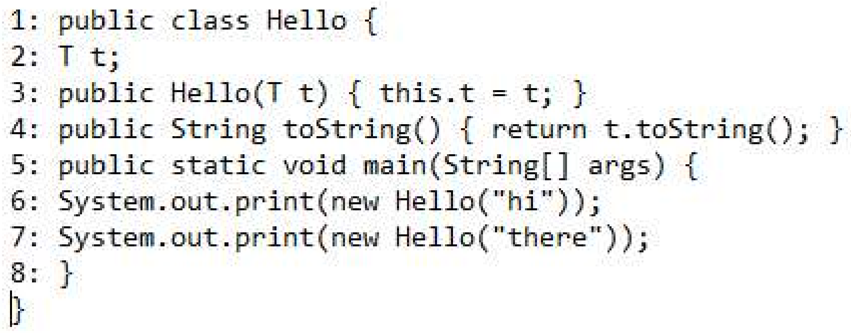
☐ C. The code will not compile because of line 6.

☐ D. The code will not compile because of line 8.

☑ E. The code will not compile because of line 9

41.What is the result of the following code? *

1/1



```
1: public class Hello {
2: T t;
3: public Hello(T t) { this.t = t; }
4: public String toString() { return t.toString(); }
5: public static void main(String[] args) {
6: System.out.print(new Hello("hi"));
7: System.out.print(new Hello("there"));
8: }
}
```

☐ A. hi

☐ B. hi followed by a runtime exception

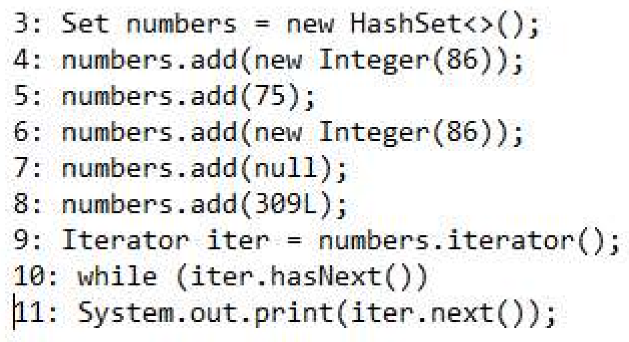
☑ C. hithere

☐ D. Compiler error on line 4

☐ E. Compiler error on line 6

☐ F. Compiler error on line 7

42.Which of the following statements are true? (Select two.) * 1/1



```
3: Set numbers = new HashSet<>();
4: numbers.add(new Integer(86));
5: numbers.add(75);
6: numbers.add(new Integer(86));
7: numbers.add(null);
8: numbers.add(309L);
9: Iterator iter = numbers.iterator();
10: while (iter.hasNext())
11: System.out.print(iter.next());
```

☑ A. The code compiles successfully.

□ B. The output is 8675null309.

□ C. The output is 867586null309.

☑ D. The output is indeterminate.

□ E. There is a compiler error on line 3.

□ F. There is a compiler error on line 9.

□ G. An exception is thrown.

43.What is the result of the following code? TreeSet tree = new TreeSet(); tree.add("one"); tree.add("One"); tree.add("ONE"); System.out.println(tree.ceiling("On")); *1/1

☐ A. On

☐ B. one

☑ C. One

☐ D. ONE

☐ E. The code does not compile.

☐ F. An exception is thrown.

44.Which of the answer choices are valid given the following declaration? Map map = new HashMap<>(); *1/1

☐ A. map.add("pi", 3.14159);

☐ B. map.add("e", 2L);

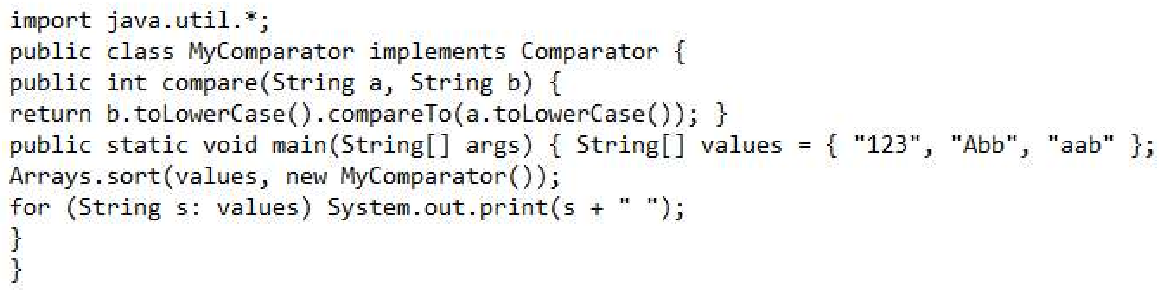
☐ C. map.add("log(1)", new Double(0.0));

☐ D. map.add('x', new Double(123.4));

☑ E. None of the above

45.What is the result of the following program? *

1/1



```
import java.util.*;
public class MyComparator implements Comparator {
public int compare(String a, String b) {
return b.toLowerCase().compareTo(a.toLowerCase()); }
public static void main(String[] args) { String[] values = { "123", "Abb", "aab" };
Arrays.sort(values, new MyComparator());
for (String s: values) System.out.print(s + " ");
}
}
```

☑ A. Abb aab 123

□ B. aab Abb 123

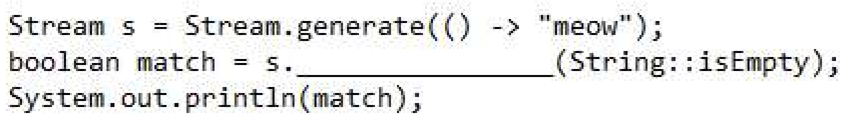
□ C. 123 Abb aab

□ D. 123 aab Abb

□ E. The code does not compile.

□ F. A runtime exception is thrown.

46.Which of the following can fill in the blank so that the code prints out false? (Choose all that apply.) *1/1



```
Stream s = Stream.generate(() -> "meow");
boolean match = s.________________(String::isEmpty);
System.out.println(match);
```

☑ A. allMatch

☐ B. anyMatch

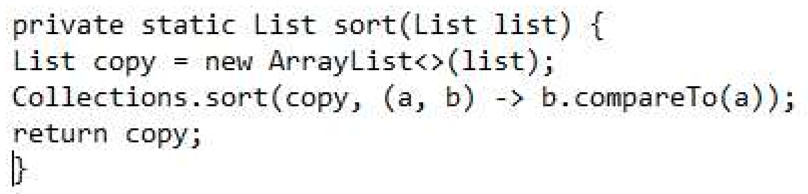
☐ C. findAny

☐ D. findFirst

☐ E. noneMatch

☐ F. None of the above

47.We have a method that returns a sorted list without changing the original. Which of the following can replace the method implementation to do the same with streams? *1/1



```
private static List sort(List list) {
List copy = new ArrayList<>(list);
Collections.sort(copy, (a, b) -> b.compareTo(a));
return copy;
}
```

□ A. return list.stream() .compare((a, b) -> b.compareTo(a)) .collect(Collectors.toList());

□ B. return list.stream() .compare((a, b) -> b.compareTo(a)) .sort();

□ C. return list.stream() .compareTo((a, b) -> b.compareTo(a)) .collect(Collectors.toList());

□ D. return list.stream() .compareTo((a, b) -> b.compareTo(a)) .sort();

□ E. return list.stream() .sorted((a, b) -> b.compareTo(a)) .collect();

☑ F. return list.stream() .sorted((a, b) -> b.compareTo(a)) .collect(Collectors.toList());

48.Which of the following are true given the declaration IntStream is = IntStream. empty()? (Choose all that apply.) *1/1

☐ A. is.average() returns the type int.

☐ B. is.average() returns the type OptionalInt.

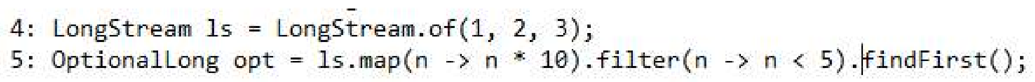
☐ C. is.findAny() returns the type int.

☑ D. is.findAny() returns the type OptionalInt.

☑ E. is.sum() returns the type int.

☐ F. is.sum() returns the type OptionalInt.

49.Which of the following can we add after line 5 for the code to run without error and not produce any output? (Choose all that apply.) *1/1



```
4: LongStream ls = LongStream.of(1, 2, 3);
5: OptionalLong opt = ls.map(n -> n * 10).filter(n -> n < 5).findFirst();
```

☐ A. if (opt.isPresent()) System.out.println(opt.get());

☑ B. if (opt.isPresent()) System.out.println(opt.getAsLong());

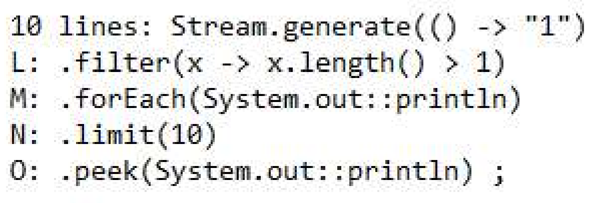
☐ C. opt.ifPresent(System.out.println)

☑ D. opt.ifPresent(System.out::println)

☐ E. None of these; the code does not compile.

☐ F. None of these; line 5 throws an exception at runtime.

50.Select from the following statements and indicate the order in which they would appear to output *0/1



```
10 lines: Stream.generate(() -> "1")
L: .filter(x -> x.length() > 1)
M: .forEach(System.out::println)
N: .limit(10)
O: .peek(System.out::println) ;
```

□ A. L, N

□ B. L, N, O

□ C. L, N, M

☑ D. L, N, M, O

□ E. L, O, M

□ F. N, M

□ G. N, O