Examen de Certificación VII

1. Which of the following are true about encapsulation? (Choose all that apply.)
   A. It allows getters.
   B. It allows setters.
   C. It requires specific naming conventions.
   D. It uses package-private instance variables.
   E. It uses private instance variables.

2. What is the output of the following code?

```
import rope.*;
import static rope.Rope.*;
public class RopeSwing {
      private static Rope rope1 = new Rope();
      private static Rope rope2 = new Rope();
      {
            System.out.println(rope1.length);
      }
      public static void main(String[] args) {
            rope1.length = 2;
            rope2.length = 8;
            System.out.println(rope1.length);
      }
}
package rope;
public class Rope {
      public static int length = 0;
}
```

   A. 02
   B. 08
   C. 2
   D. 8
   E. The code does not compile.
   F. An exception is thrown.

3. Which of the following can replace line 2 to make this code compile? (Choose all that apply.)

```
1: import java.util.*;
2: // INSERT CODE HERE
3: public class Imports {
4: public void method(ArrayList<String> list) {
5: sort(list);
6: }
7: }
```

   A. import static java.util.Collections;
   B. import static java.util.Collections.*;
   C. import static java.util.Collections.sort(ArrayList<String>);
   D. static import java.util.Collections;
   E. static import java.util.Collections.*;
   F. static import java.util.Collections.sort(ArrayList<String>);

4. What is the output of the following code?

```
1: class Arthropod {
2: protected void printName(long input) {
3: System.out.print("Arthropod");
```

```
4: }
5: void printName(int input) {
6: System.out.print("Spooky");
7: } }
8: public class Spider extends Arthropod {
9: protected void printName(int input) {
10: System.out.print("Spider");
11: }
12: public static void main(String[] args) {
13: Arthropod a = new Spider();
14: a.printName((short)4);
15: a.printName(4);
16: a.printName(5L);
17: } }
```

A. SpiderSpiderArthropod
B. SpiderSpiderSpider
C. SpiderSpookyArthropod
D. SpookySpiderArthropod
E. The code will not compile because of line 5.
F. The code will not compile because of line 9.
G. None of the above

5. Which of these classes compile and will include a default constructor created by the compiler? (Choose all that apply.)

A. `public class Bird {}`
B.
```
public class Bird {
public bird() {}
}
```
C.
```
public class Bird {
public bird(String name) {}
}
```
D.
```
public class Bird {
public Bird() {}
}
```
E.
```
public class Bird {
Bird(String name) {}
}
```
F.
```
public class Bird {
private Bird(int age) {}
}
```
G.
```
public class Bird {
public Bird bird() {return null;}
}
```

6. What is the output of the following code?

```
1: class Reptile {
2: {System.out.print("A");}
3: public Reptile(int hatch) {}
4: void layEggs() {
5: System.out.print("Reptile");
6: } }
7: public class Lizard extends Reptile {
8: static {System.out.print("B");}
9: public Lizard(int hatch) {}
10: public final void layEggs() {
11: System.out.print("Lizard");
```

```
12: }
13: public static void main(String[] args) {
14: Reptile reptile = new Lizard(1);
15: reptile.layEggs();
16: } }
```

A. AALizard
B. BALizard
C. BLizardA
D. ALizard
E. The code will not compile because of line 10.
F. None of the above

7. Given:

```
public abstract interface Frobnicate { public void
twiddle(String s); }
```

Which is a correct class? (Choose all that apply.)

A.
```
public abstract class Frob implements Frobnicate {
public abstract void twiddle(String s) { }
}
```

B.
```
public abstract class Frob implements Frobnicate { }
```

C.
```
public class Frob extends Frobnicate {
public void twiddle(Integer i) { }
}
```

D.
```
public class Frob implements Frobnicate {
public void twiddle(Integer i) { }
}
```

E.
```
public class Frob implements Frobnicate {
public void twiddle(String i) { }
public void twiddle(Integer s) { }
}
```

8. Given:

```
class Top {
public Top(String s) { System.out.print("B"); }
}
public class Bottom2 extends Top {
public Bottom2(String s) { System.out.print("D"); }
public static void main(String [] args) {
new Bottom2("C");
System.out.println(" ");
}
}
```

What is the result?

A. BD
B. DB
C. BDC
D. DBC
E. Compilation fails

9. Given:

```
class Bird {
{ System.out.print("b1 "); }
public Bird() { System.out.print("b2 "); }
}
class Raptor extends Bird {
static { System.out.print("r1 "); }
public Raptor() { System.out.print("r2 "); }
```

```
{ System.out.print("r3 "); }
static { System.out.print("r4 "); }
}
class Hawk extends Raptor {
public static void main(String[] args) {
System.out.print("pre ");
new Hawk();
System.out.println("hawk ");
}
}
```

What is the result?

A. pre b1 b2 r3 r2 hawk
B. pre b2 b1 r2 r3 hawk
C. pre b2 b1 r2 r3 hawk r1 r4
D. r1 r4 pre b1 b2 r3 r2 hawk
E. r1 r4 pre b2 b1 r2 r3 hawk
F. pre r1 r4 b1 b2 r3 r2 hawk
G. pre r1 r4 b2 b1 r2 r3 hawk
H. The order of output cannot be predicted
I. Compilation fails

10. Given:

```
public class Locomotive {
Locomotive() { main("hi"); }
public static void main(String[] args) {
System.out.print("2 ");
}
public static void main(String args) {
System.out.print("3 " + args);
}
}
```

What is the result? (Choose all that apply.)

A. 2 will be included in the output
B. 3 will be included in the output
C. hi will be included in the output
D. Compilation fails
E. An exception is thrown at runtime

11. Given:

```
3. class Dog {
4. public void bark() { System.out.print("woof "); }
5. }
6. class Hound extends Dog {
7. public void sniff() { System.out.print("sniff "); }
8. public void bark() { System.out.print("howl "); }
9. }
10. public class DogShow {
11. public static void main(String[] args) { new DogShow().go();
}
12. void go() {
13. new Hound().bark();
14. ((Dog) new Hound()).bark();
15. ((Dog) new Hound()).sniff();
16. }
17. }
```

What is the result? (Choose all that apply.)

A. howl howl sniff

B. howl woof sniff
C. howl howl followed by an exception
D. howl woof followed by an exception
E. Compilation fails with an error at line 14
F. Compilation fails with an error at line 15

12. Given:

```
3. class Mammal {
4. String name = "furry ";
5. String makeNoise() { return "generic noise"; }
6. }
7. class Zebra extends Mammal {
8. String name = "stripes ";
9. String makeNoise() { return "bray"; }
10. }
11. public class ZooKeeper {
12. public static void main(String[] args) { new
ZooKeeper().go(); }
13. void go() {
14. Mammal m = new Zebra();
15. System.out.println(m.name + m.makeNoise());
16. }
17. }
```

What is the result?

A. furry bray
B. stripes bray
C. furry generic noise
D. stripes generic noise
E. Compilation fails
F. An exception is thrown at runtime

13. Given:

```
3. public class Tenor extends Singer {
4. public static String sing() { return "fa"; }
5. public static void main(String[] args) {
6. Tenor t = new Tenor();
7. Singer s = new Tenor();
8. System.out.println(t.sing() + " " + s.sing());
9. }
10. }
11. class Singer { public static String sing() { return "la"; }
}
```

What is the result?

A. fa fa
B. fa la
C. la la
D. Compilation fails
E. An exception is thrown at runtime