1. Which of the following are valid record declarations? (Choose all that apply.)

```
public record Iguana(int age) {
  private static final int age = 10; }

public final record Gecko() {}

public abstract record Chameleon()  {
  private static String name; }

public record BeardedDragon(boolean fun) {
  @Override public boolean fun() { return false; } }

public record Newt(long size) {
  @Override public boolean equals(Object obj) { return false; }
  public void setSize(long size) {
    this.size = size;
  } }
```

A. Iguana

B. Gecko

C. Chameleon

D. BeardedDragon

E. Newt

F. None of the above

2. Which of the following statements can be inserted in the blank line so that the code will compile successfully? (Choose all that apply.)

```
interface CanHop {}
public class Frog implements CanHop {
  public static void main(String[] args) {
    _____ frog = new TurtleFrog();
  }
}
class BrazilianHornedFrog extends Frog {}
class TurtleFrog extends Frog {}
```

A. Frog

B. TurtleFrog

C. BrazilianHornedFrog

D. CanHop

E. var

F. Long

G. None of the above; the code contains a compilation error.

3. What is the result of the following program?

```
public class Favorites {
  enum Flavors {
    VANILLA, CHOCOLATE, STRAWBERRY
    static final Flavors DEFAULT = STRAWBERRY;
  }
  public static void main(String[] args) {
    for(final var e : Flavors.values())
      System.out.print(e.ordinal()+" ");
  }
}
```

A. 0 1 2

B. 1 2 3

C. Exactly one line of code does not compile.

D. More than one line of code does not compile.

E. The code compiles but produces an exception at runtime.

F. None of the above

4. What is the output of the following program?

```
public sealed class ArmoredAnimal permits Armadillo {
  public ArmoredAnimal(int size) {}
  @Override public String toString() { return "Strong"; }
  public static void main(String[] a) {
    var c = new Armadillo(10, null);
    System.out.println(c);
  }
}
class Armadillo extends ArmoredAnimal {
  @Override public String toString() { return "Cute"; }
  public Armadillo(int size, String name) {
    super(size);
  }
}
```

A. Strong

B. Cute

C. The program does not compile.

D. The code compiles but produces an exception at runtime.

E. None of the above

5. Which statements about the following program are correct? (Choose all that apply.)

```
1: interface HasExoskeleton {
2:    double size = 2.0f;
3:    abstract int getNumberOfSections();
4: }
5: abstract class Insect implements HasExoskeleton {
6:    abstract int getNumberOfLegs();
7: }
8: public class Beetle extends Insect {
9:    int getNumberOfLegs() { return 6; }
10:   int getNumberOfSections(int count) { return 1; }
11: }
```

A. It compiles without issue.

B. The code will produce a ClassCastException if called at runtime.

C. The code will not compile because of line 2.

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

E. The code will not compile because of line 8.

F. The code will not compile because of line 10.

6. Which statements about the following program are correct? (Choose all that apply.)

```
1: public abstract interface Herbivore {
2:   int amount = 10;
3:   public void eatGrass();
4:   public abstract int chew() { return 13; }
5: }
6:
7: abstract class IsAPlant extends Herbivore {
8:   Object eatGrass(int season) { return null; }
9: }
```

A. It compiles and runs without issue.

B. The code will not compile because of line 1.

C. The code will not compile because of line 2.

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

E. The code will not compile because of line 7.

F. The code will not compile because line 8 contains an invalid method override.

7. What is the output of the following program?

```
1: interface Aquatic {
2:   int getNumOfGills(int p);
3: }
4: public class ClownFish implements Aquatic {
5:   String getNumOfGills() { return "14"; }
6:   int getNumOfGills(int input) { return 15; }
7:   public static void main(String[] args) {
8:     System.out.println(new ClownFish().getNumOfGills(-1));
9: } }
```

A. 14

B. 15

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 line 6.

F. None of the above

8. When inserted in order, which modifiers can fill in the blank to create a properly encapsulated class? (Choose all that apply.)

```
public class Rabbits {
  _______ int numRabbits = 0;
  _______ void multiply() {
   numRabbits *= 6;
  }
  _______ int getNumberOfRabbits() {
   return numRabbits;
  }
}
```

A. private, public, and public

B. private, protected, and private

C. private, private, and protected

D. public, public, and public

E. The class cannot be properly encapsulated since multiply() does not begin with set.

F. None of the above

9. Which of the following statements can be inserted in the blank so that the code will compile successfully? (Choose all that apply.)

```
abstract class Snake {}
class Cobra extends Snake {}
class GardenSnake extends Cobra {}
public class SnakeHandler {
  private Snake snakey;
  public void setSnake(Snake mySnake) { this.snakey = mySnake; }
  public static void main(String[] args) {
    new SnakeHandler().setSnake(_____________________);
  }
}
```

A. new Cobra()

B. new Snake()

C. new Object()

D. new String("Snake")

E. new GardenSnake()

F. null

G. None of the above. The class does not compile, regardless of the value inserted in the blank.

10. What types can be inserted in the blanks on the lines marked X and Z that allow the code to compile? (Choose all that apply.)

```
interface Walk { private static List move() { return null; } }
interface Run extends Walk { public ArrayList move(); }
class Leopard implements Walk {
  public _________ move() {  // X
    return null;
  }
}
class Panther implements Run {
  public _________ move() {  // Z
    return null;
  }
}
```

A. Integer on the line marked X

B. ArrayList on the line marked X

C. List on the line marked X

D. List on the line marked Z

E. ArrayList on the line marked Z

F. None of the above, since the Run interface does not compile

G. The code does not compile for a different reason.

11. What is the result of the following code? (Choose all that apply.)

```
1:  public class Movie {
2:    private int butter = 5;
3:    private Movie() {}
4:    protected class Popcorn {
5:      private Popcorn() {}
6:      public static int butter = 10;
7:      public void startMovie() {
8:        System.out.println(butter);
9:      }
10:   }
11:   public static void main(String[] args) {
12:     var movie = new Movie();
13:     Movie.Popcorn in = new Movie().new Popcorn();
14:     in.startMovie();
15:   }}
```

A. The output is 5.

B. The output is 10.

C. Line 6 generates a compiler error.

D. Line 12 generates a compiler error.

E. Line 13 generates a compiler error.

F. The code compiles but produces an exception at runtime.

12. 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 requires public instance variables.

E. It requires private instance variables.

13. What is the result of the following program?

```
public class Weather {
  enum Seasons {
    WINTER, SPRING, SUMMER, FALL
  }

  public static void main(String[] args) {
    Seasons v = null;
    switch (v) {
      case Seasons.SPRING -> System.out.print("s");
      case Seasons.WINTER -> System.out.print("w");
      case Seasons.SUMMER -> System.out.print("m");
      default -> System.out.println("missing data"); }
  }
}
```

A. s

B. w

C. m

D. missing data

E. Exactly one line of code does not compile.

F. More than one line of code does not compile.

G. The code compiles but produces an exception at runtime.

14. Which statements about sealed classes are correct? (Choose all that apply.)

A. A sealed interface restricts which subinterfaces may extend it.

B. A sealed class cannot be indirectly extended by a class that is not listed in its permits clause.

C. A sealed class can be extended by an abstract class.

D. A sealed class can be extended by a subclass that uses the non-sealed modifier.

E. A sealed interface restricts which subclasses may implement it.

F. A sealed class cannot contain any nested subclasses.

G. None of the above

15. Which lines, when entered independently into the blank, allow the code to print Not scared at runtime? (Choose all that apply.)

```
public class Ghost {
  public static void boo() {
    System.out.println("Not scared");
  }
```

```
  protected final class Spirit {
    public void boo() {
      System.out.println("Booo!!!");
    }
  }
  public static void main(String... haunt) {
    var g = new Ghost().new Spirit() {};
    _____;
  }
}
```

A. g.boo()

B. g.super.boo()

C. new Ghost().boo()

D. g.Ghost.boo()

E. new Spirit().boo()

F. Ghost.boo()

G. None of the above

16. The following code appears in a file named Ostrich.java. What is the result of compiling the source file?

```
1: public class Ostrich {
2:   private int count;
3:   static class OstrichWrangler {
4:     public int stampede() {
5:       return count;
6:     } } }
```

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

B. The code compiles successfully, and two bytecode files are generated: Ostrich.class and OstrichWrangler.class.

C. The code compiles successfully, and two bytecode files are generated: Ostrich.class and Ostrich$OstrichWrangler.class.

D. A compiler error occurs on line 3.

E. A compiler error occurs on line 5.

17. Which lines of the following interface declarations do not compile? (Choose all that apply.)

```
1: public interface Omnivore {
2:   int amount = 10;
3:   static boolean gather = true;
```

```
4:  static void eatGrass() {}
5:  int findMore() { return 2; }
6:  default float rest() { return 2; }
7:  protected int chew() { return 13; }
8:  private static void eatLeaves() {}
9: }
```

A. All of the lines compile without issue.

B. Line 2

C. Line 3

D. Line 4

E. Line 5

F. Line 6

G. Line 7

H. Line 8

18. What is printed by the following program?

```
public class Deer {
  enum Food {APPLES, BERRIES, GRASS}
  protected class Diet {
    private Food getFavorite() {
      return Food.BERRIES;
    }
  }
  public static void main(String[] seasons) {
    System.out.print(switch(new Diet().getFavorite()) {
      case APPLES -> "a";
      case BERRIES -> "b";
      default -> "c";
    });
  } }
```

A. a

B. b

C. c

D. The code declaration of the Diet class does not compile.

E. The main() method does not compile.

F. The code compiles but produces an exception at runtime.

G. None of the above

19. Which of the following are printed by the Bear program? (Choose all that apply.)

```
public class Bear {
  enum FOOD {
    BERRIES, INSECTS {
      public boolean isHealthy() { return true; }},
    FISH, ROOTS, COOKIES, HONEY;
    public abstract boolean isHealthy();
  }
  public static void main(String[] args) {
    System.out.print(FOOD.INSECTS);
    System.out.print(FOOD.INSECTS.ordinal());
    System.out.print(FOOD.INSECTS.isHealthy());
    System.out.print(FOOD.COOKIES.isHealthy());
  }
}
```

A. insects

B. INSECTS

C. 0

D. 1

E. false

F. true

G. The code does not compile.

20. Which statements about polymorphism and method inheritance are correct? (Choose all that apply.)

A. Given an arbitrary instance of a class, it cannot be determined until runtime which overridden method will be executed in a parent class.

B. It cannot be determined until runtime which hidden method will be executed in a parent class.

C. Marking a method static prevents it from being overridden or hidden.

D. Marking a method final prevents it from being overridden or hidden.

E. The reference type of the variable determines which overridden method will be called at runtime.

F. The reference type of the variable determines which hidden method will be called at runtime.

21. Given the following record declaration, which lines of code can fill in the blank and allow the code to compile? (Choose all that apply.)

```
public record RabbitFood(int size, String brand, LocalDate expires) {
  public static int MAX_STORAGE = 100;
  public RabbitFood() {
    _____;
  }
}
```

A. size = MAX_STORAGE

B. this.size = 10

C. if(expires.isAfter(LocalDate.now())) throw new RuntimeException()

D. if(brand==null) super.brand = "Unknown"

E. throw new RuntimeException()

F. None of the above

22. Which of the following can be inserted in the rest() method? (Choose all that apply.)

```
public class Lion {
  class Cub {}
  static class Den {}
  static void rest() {
    _____;
  }}
```

A. Cub a = Lion.new Cub()

B. Lion.Cub b = new Lion().Cub()

C. Lion.Cub c = new Lion().new Cub()

D. var d = new Den()

E. var e = Lion.new Cub()

F. Lion.Den f = Lion.new Den()

G. Lion.Den g = new Lion.Den()

H. var h = new Cub()

23. Given the following program, what can be inserted into the blank line that would allow it to print Swim! at runtime?

```
interface Swim {
  default void perform() { System.out.print("Swim!"); }
}
interface Dance {
  default void perform() { System.out.print("Dance!"); }
}
```

```
public class Penguin implements Swim, Dance {
  public void perform() { System.out.print("Smile!"); }
  private void doShow() {
    _____;
  }
  public static void main(String[] eggs) {
    new Penguin().doShow();
  }
}
```

A. super.perform()

B. Swim.perform()

C. super.Swim.perform()

D. Swim.super.perform()

E. The code does not compile regardless of what is inserted into the blank.

F. The code compiles, but due to polymorphism, it is not possible to produce the requested output without creating a new object.

24. Which lines of the following interface do not compile? (Choose all that apply.)

```
1: public interface BigCat {
2:   abstract String getName();
3:   static int hunt() { getName(); return 5; }
4:   default void climb() { rest(); }
5:   private void roar() { getName();  climb(); hunt(); }
6:   private static boolean sneak() { roar(); return true; }
7:   private int rest() { return 2; };
8: }
```

A. Line 2

B. Line 3

C. Line 4

D. Line 5

E. Line 6

F. Line 7

G. None of the above

25. What does the following program print?

```
1: public class Zebra {
2:   private int x = 24;
```

```
3:  public int hunt() {
4:    String message = "x is ";
5:    abstract class Stripes {
6:      private int x = 0;
7:      public void print() {
8:        System.out.print(message + Zebra.this.x);
9:      }
10:   }
11:   var s = new Stripes() {};
12:   s.print();
13:   return x;
14: }
15: public static void main(String[] args) {
16:   new Zebra().hunt();
17: } }
```

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. None of the above

26. Which statements about the following enum are true? (Choose all that apply.)

```
1: public enum Animals {
2:   MAMMAL(true), INVERTEBRATE(Boolean.FALSE), BIRD(false),
3:   REPTILE(false), AMPHIBIAN(false), FISH(false) {
4:     public int swim() { return 4; }
5:   }
6:   final boolean hasHair;
7:   public Animals(boolean hasHair) {
8:     this.hasHair = hasHair;
9:   }
10:  public boolean hasHair() { return hasHair; }
11:  public int swim() { return 0; }
12: }
```

A. Compiler error on line 2

B. Compiler error on line 3

C. Compiler error on line 7

D. Compiler error on line 8

E. Compiler error on line 10

F. Compiler error on another line

G. The code compiles successfully.

27. Assuming a record is defined with at least one field, which components does the compiler always insert, each of which may be overridden or redeclared? (Choose all that apply.)

A. A no-argument constructor

B. An accessor method for each field

C. The toString() method

D. The equals() method

E. A mutator method for each field

F. A sort method for each field

G. The hashCode() method

28. Which of the following classes and interfaces do not compile? (Choose all that apply.)

```
public abstract class Camel { void travel(); }

public interface EatsGrass { private abstract int chew(); }

public abstract class Elephant {
  abstract private class SleepsAlot {
    abstract int sleep();
  } }

public class Eagle { abstract soar(); }

public interface Spider { default void crawl() {} }
```

A. Camel

B. EatsGrass

C. Elephant

D. Eagle

E. Spider

F. None of the classes or interfaces compile.

29. How many lines of the following program contain a compilation error?

```
1: class Primate {
2:    protected int age = 2;
```

```
3:   { age = 1; }
4:   public Primate() {
5:     this().age = 3;
6:   }
7: }
8: public class Orangutan {
9:   protected int age = 4;
10:  { age = 5; }
11:  public Orangutan() {
12:    this().age = 6;
13:  }
14:  public static void main(String[] bananas) {
15:    final Primate x = (Primate)new Orangutan();
16:    System.out.println(x.age);
17:  }
18: }
```

A. None, and the program prints 1 at runtime.

B. None, and the program prints 3 at runtime.

C. None, but it causes a ClassCastException at runtime.

D. 1

E. 2

F. 3

G. 4

30. Assuming the following classes are declared as top-level types in the same file, which classes contain compiler errors? (Choose all that apply.)

```
sealed class Bird {
  public final class Flamingo extends Bird {}
}

sealed class Monkey {}

class EmperorTamarin extends Monkey {}

non-sealed class Mandrill extends Monkey {}

sealed class Friendly extends Mandrill permits Silly {}

final class Silly {}
```

A. Bird

B. Monkey

C. EmperorTamarin

D. Mandrill

E. Friendly

F. Silly

G. All of the classes compile without issue.