

Review Questions

1. Which of the following can fill in the blank in this code to make it compile? (Choose all that apply)

```
public class Ant {  
    _____ void method() { }  
}
```

- A. default
 - B. final
 - C. private
 - D. Public
 - E. String
 - F. zzz:
2. Which of the following compile? (Choose all that apply)
- A. `final static void method4() { }`
 - B. `public final int void method() { }`
 - C. `private void int method() { }`
 - D. `static final void method3() { }`
 - E. `void final method() {}`
 - F. `void public method() { }`
3. Which of the following methods compile? (Choose all that apply)
- A. `public void methodA() { return;}`
 - B. `public void methodB() { return null;}`
 - C. `public void methodD() {}`
 - D. `public int methodD() { return 9;}`
 - E. `public int methodE() { return 9.0;}`
 - F. `public int methodF() { return;}`
 - G. `public int methodG() { return null;}`
4. Which of the following compile? (Choose all that apply)
- A. `public void moreA(int... nums) {}`
 - B. `public void moreB(String values, int... nums) {}`
 - C. `public void moreC(int... nums, String values) {}`
 - D. `public void moreD(String... values, int... nums) {}`
 - E. `public void moreE(String[] values, ...int nums) {}`
 - F. `public void moreF(String... values, int[] nums) {}`
 - G. `public void moreG(String[] values, int[] nums) {}`

5. Given the following method, which of the method calls return 2? (Choose all that apply)
- ```
public int howMany(boolean b, boolean... b2) {
 return b2.length;
}
```
- A. `howMany();`
  - B. `howMany(true);`
  - C. `howMany(true, true);`
  - D. `howMany(true, true, true);`
  - E. `howMany(true, {true});`
  - F. `howMany(true, {true, true});`
  - G. `howMany(true, new boolean[2]);`
6. Which of the following are true? (Choose all that apply)
- 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`.
7. 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)
- ```
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
8. Which of the following are true? (Choose all that apply)
- 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.
9. Which are methods using JavaBeans naming conventions for accessors and mutators? (Choose all that apply)
- 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;}`
10. What is the output of the following code?
- ```
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.

11. Which are true of the following code? (Choose all that apply)

```
1: public class Rope {
2:     public static void swing() {
3:         System.out.print("swing ");
4:     }
5:     public void climb() {
6:         System.out.println("climb ");
7:     }
8:     public static void play() {
9:         swing();
10:        climb();
11:    }
12:    public static void main(String[] args) {
13:        Rope rope = new Rope();
14:        rope.play();
15:        Rope rope2 = null;
16:        rope2.play();
17:    }
18: }
```

- A. The code compiles as is.
- B. There is exactly one compiler error in the code.
- C. There are exactly two compiler errors in the code.
- D. If the lines with compiler errors are removed, the output is `climb climb`.
- E. If the lines with compiler errors are removed, the output is `swing swing`.
- F. If the lines with compile errors are removed, the code throws a `NullPointerException`.

12. 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.

13. How many compiler errors are in the following code?

```
1: public class RopeSwing {
2:     private static final String leftRope;
3:     private static final String rightRope;
4:     private static final String bench;
5:     private static final String name = "name";
```

```
6:  static {
7:      leftRope = "left";
8:      rightRope = "right";
9:  }
10: static {
11:     name = "name";
12:     rightRope = "right";
13: }
14: public static void main(String[] args) {
15:     bench = "bench";
16: }
17: }
```

- A. 0
- B. 1
- C. 2
- D. 3
- E. 4
- F. 5

14. 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>);`

15. What is the result of the following statements?

```
1: public class Test {
2:     public void print(byte x) {
3:         System.out.print("byte");
4:     }
5:     public void print(int x) {
6:         System.out.print("int");
7:     }
8: }
```

```
7:    }
8:    public void print(float x) {
9:        System.out.print("float");
10:    }
11:    public void print(Object x) {
12:        System.out.print("Object");
13:    }
14:    public static void main(String[] args) {
15:        Test t = new Test();
16:        short s = 123;
17:        t.print(s);
18:        t.print(true);
19:        t.print(6.789);
20:    }
21: }
```

- A. `bytefloatObject`
- B. `intfloatObject`
- C. `byteObjectfloat`
- D. `intObjectfloat`
- E. `intObjectObject`
- F. `byteObjectObject`

16. What is the result of the following program?

```
1: public class Squares {
2:     public static long square(int x) {
3:         long y = x * (long) x;
4:         x = -1;
5:         return y;
6:     }
7:     public static void main(String[] args) {
8:         int value = 9;
9:         long result = square(value);
10:        System.out.println(value);
11:    } }
```

- A. -1
- B. 9
- C. 81
- D. Compiler error on line 9.
- E. Compiler error on a different line.

17. Which of the following are output by the following code? (Choose all that apply)

```
public class StringBuilders {  
    public static StringBuilder work(StringBuilder a,  
    StringBuilder b) {  
        a = new StringBuilder("a");  
        b.append("b");  
        return a;  
    }  
    public static void main(String[] args) {  
        StringBuilder s1 = new StringBuilder("s1");  
        StringBuilder s2 = new StringBuilder("s2");  
        StringBuilder s3 = work(s1, s2);  
        System.out.println("s1 = " + s1);  
        System.out.println("s2 = " + s2);  
        System.out.println("s3 = " + s3);  
    }  
}
```

- A. s1 = a
 - B. s1 = s1
 - C. s2 = s2
 - D. s2 = s2b
 - E. s3 = a
 - F. s3 = null
 - G. The code does not compile.
18. Which of the following are true? (Choose 2)
- A. this() can be called from anywhere in a constructor.
 - B. this() can be called from any instance method in the class.
 - C. this.variableName can be called from any instance method in the class.
 - D. this.variableName can be called from any static method in the class.
 - E. You must include a default constructor in the code if the compiler does not include one.
 - F. You can call the default constructor written by the compiler using this().
 - G. You can access a private constructor with the main() method.
19. Which of these classes compile and use a default constructor? (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 { void Bird() {} }`

20. Which code can be inserted to have the code print 2?

```
public class BirdSeed {
    private int numberBags;
    boolean call;

    public BirdSeed() {
        // LINE 1
        call = false;
        // LINE 2
    }
    public BirdSeed(int numberBags) {
        this.numberBags = numberBags;
    }
    public static void main(String[] args) {
        BirdSeed seed = new BirdSeed();
        System.out.println(seed.numberBags);
    } }
```

- A. Replace line 1 with `BirdSeed(2);`
- B. Replace line 2 with `BirdSeed(2);`
- C. Replace line 1 with `new BirdSeed(2);`
- D. Replace line 2 with `new BirdSeed(2);`
- E. Replace line 1 with `this(2);`
- F. Replace line 2 with `this(2);`

21. Which of the following complete the constructor so that this code prints out 50? (Choose all that apply)

```
public class Cheetah {
    int numSpots;
    public Cheetah(int numSpots) {
        // INSERT CODE HERE
    }
    public static void main(String[] args) {
        System.out.println(new Cheetah(50).numSpots);
    }
}
```

- A. numSpots = numSpots;
- B. numSpots = this.numSpots;
- C. this.numSpots = numSpots;
- D. numSpots = super.numSpots;
- E. super.numSpots = numSpots;
- F. None of the above.

22. What is the result of the following?

```
1: public class Order {  
2:     static String result = "";  
3:     { result += "c"; }  
4:     static  
5:     { result += "u"; }  
6:     { result += "r"; }  
7: }
```

```
1: public class OrderDriver {  
2:     public static void main(String[] args) {  
3:         System.out.print(Order.result + " ");  
4:         System.out.print(Order.result + " ");  
5:         new Order();  
6:         new Order();  
7:         System.out.print(Order.result + " ");  
8:     }  
9: }
```

- A. curur
- B. ucr cr
- C. u ucr cr
- D. u u cur cur
- E. u u ucr cr
- F. ur ur urc
- G. The code does not compile.

23. What is the result of the following?

```
1: public class Order {  
2:     String value = "t";  
3:     { value += "a"; }  
4:     { value += "c"; }  
5:     public Order() {
```

```
6:     value += "b";
7: }
8: public Order(String s) {
9:     value += s;
10: }
11: public static void main(String[] args) {
12:     Order order = new Order("f");
13:     order = new Order();
14:     System.out.println(order.value);
15: } }
```

- A. tacb
- B. tacf
- C. tacbf
- D. tacfb
- E. tacftacb
- F. The code does not compile.
- G. An exception is thrown.

24. Which of the following will compile when inserted in the following code? (Choose all that apply)

```
public class Order3 {
    final String value1 = "1";
    static String value2 = "2";
    String value3 = "3";
    {
        // CODE SNIPPET 1
    }
    static {
        // CODE SNIPPET 2
    }
}
```

- A. value1 = "d"; instead of // CODE SNIPPET 1
- B. value2 = "e"; instead of // CODE SNIPPET 1
- C. value3 = "f"; instead of // CODE SNIPPET 1
- D. value1 = "g"; instead of // CODE SNIPPET 2
- E. value2 = "h"; instead of // CODE SNIPPET 2
- F. value3 = "i"; instead of // CODE SNIPPET 2

25. Which of the following are true about the following code? (Choose all that apply)

```
public class Create {  
    Create() {  
        System.out.print("1 ");  
    }  
    Create(int num) {  
        System.out.print("2 ");  
    }  
    Create(Integer num) {  
        System.out.print("3 ");  
    }  
    Create(Object num) {  
        System.out.print("4 ");  
    }  
    Create(int... nums) {  
        System.out.print("5 ");  
    }  
    public static void main(String[] args) {  
        new Create(100);  
        new Create(1000L);  
    }  
}
```

- A. The code prints out 2 4.
 - B. The code prints out 3 4.
 - C. The code prints out 4 2.
 - D. The code prints out 4 4.
 - E. The code prints 3 4 if you remove the constructor `Create(int num)`.
 - F. The code prints 4 4 if you remove the constructor `Create(int num)`.
 - G. The code prints 5 4 if you remove the constructor `Create(int num)`.
26. What is the result of the following class?

```
1: import java.util.function.*;  
2:  
3: public class Panda {  
4:     int age;  
5:     public static void main(String[] args) {  
6:         Panda p1 = new Panda();  
7:         p1.age = 1;  
8:         check(p1, p -> p.age < 5);  
    }  
}
```

```
9:  }
10:  private static void check(Panda panda, Predicate<Panda> pred) {
11:      String result = pred.test(panda) ? "match" : "not match";
12:      System.out.print(result);
13:  } }
```

- A. match
- B. not match
- C. Compiler error on line 8.
- D. Compiler error on line 10.
- E. Compiler error on line 11.
- F. A runtime exception is thrown.

27. What is the result of the following code?

```
1: interface Climb {
2:     boolean isTooHigh(int height, int limit);
3: }
4:
5: public class Climber {
6:     public static void main(String[] args) {
7:         check((h, l) -> h.append(l).isEmpty(), 5);
8:     }
9:     private static void check(Climb climb, int height) {
10:         if (climb.isTooHigh(height, 10))
11:             System.out.println("too high");
12:         else
13:             System.out.println("ok");
14:     }
15: }
```

- A. ok
- B. too high
- C. Compiler error on line 7.
- D. Compiler error on line 10.
- E. Compiler error on a different line.
- F. A runtime exception is thrown.

28. Which of the following lambda expressions can fill in the blank? (Choose all that apply)

```
List<String> list = new ArrayList<>();
list.removeIf(_____);
```

- A. `s -> s.isEmpty()`
- B. `s -> {s.isEmpty()}`
- C. `s -> {s.isEmpty();}`
- D. `s -> {return s.isEmpty();}`
- E. `String s -> s.isEmpty()`
- F. `(String s) -> s.isEmpty()`

29. Which lambda can replace the `MySecret` class to return the same value? (Choose all that apply)

```
interface Secret {  
    String magic(double d);  
}
```

```
class MySecret implements Secret {  
    public String magic(double d) {  
        return "Poof";  
    }  
}
```

- A. `caller((e) -> "Poof");`
- B. `caller((e) -> {"Poof"});`
- C. `caller((e) -> { String e = ""; "Poof" });`
- D. `caller((e) -> { String e = ""; return "Poof"; });`
- E. `caller((e) -> { String e = ""; return "Poof" });`
- F. `caller((e) -> { String f = ""; return "Poof"; });`