**Taking the Assessment Test**

Use the following assessment test to gauge your current level of skill in Java. This test is designed to highlight some topics for your strengths and weaknesses so that you know which chapters you might want to read multiple times. Even if you do well on the assessment test, you should still read the book from cover to cover, as the real exam is quite challenging.

**The Assessment Test**

1. What is the result of the following program?

1: public class MathFunctions {

2: public static void addToInt(int x, int amountToAdd) {

3: x = x + amountToAdd;

4: }

5: public static void main(String[] args) {

6: var a = 15;

7: var b = 10;

8: MathFunctions.addToInt(a, b);

9: System.out.println(a); } }

* + 10
  + 15
  + 25
  + Compiler error on line 3
  + Compiler error on line 8
  + None of the above

1. What is the output of the following program? (Choose all that apply.)

1: interface HasTail { int getTailLength(); }

2: abstract class Puma implements HasTail {

3: protected int getTailLength() { return 4; }

4: }

5: public class Cougar implements HasTail {

6: public static void main(String[] args) {

7: var puma = new Puma();

8: System.out.println(puma.getTailLength());

9: }

10: public int getTailLength(int length) { return 2; }

11: }

1. 2
2. 4
3. The code will not compile because of line 3.
4. The code will not compile because of line 5.
5. The code will not compile because of line 7.
6. The code will not compile because of line 10.
7. The output cannot be determined from the code provided.
8. What is the output of the following code snippet?

int moon = 9, star = 2 + 2 \* 3;

float sun = star>10 ? 1 : 3;

double jupiter = (sun + moon) - 1.0f;

int mars = --moon <= 8 ? 2 : 3;

System.out.println(sun+"-"+jupiter+"-"+mars);

* 1. 1-11-2
  2. 3.0-11.0-2
  3. 1.0-11.0-3
  4. 3.0-13.0-3
  5. 3.0f-12-2
  6. The code does not compile because one of assignments requires an explicit numeric cast.

1. How many times is the word true printed?

var s1 = "Java";

var s2 = "Java";

var s3 = "Ja".concat("va");

var s4 = s3.intern();

var sb1 = new StringBuilder();

sb1.append("Ja").append("va");

System.out.println(s1 == s2);

System.out.println(s1.equals(s2));

System.out.println(s1 == s3);

System.out.println(s1 == s4);

System.out.println(sb1.toString() == s1);

System.out.println(sb1.toString().equals(s1));

* 1. Once
  2. Twice
  3. Three times
  4. Four times
  5. Five times
  6. Six times
  7. The code does not compile.

1. The following code appears in a file named Flight.java. What is the result of compiling this source file?

1: public class Flight {

2: private FlightNumber number;

3:

4: public Flight(FlightNumber number) {

5: this.number = number;

6: } }

7: public class FlightNumber {

8: public int value;

9: public String code; }

* 1. The code compiles successfully and two bytecode files are generated: Flight.class and FlightNumber.class.
  2. The code compiles successfully and one bytecode file is generated: Flight.class.
  3. A compiler error occurs on line 2.
  4. A compiler error occurs on line 4.
  5. A compiler error occurs on line 7.

1. Which of the following will run a modular program?
   1. java -cp modules mod/class
   2. java -cp modules -m mod/class
   3. java -cp modules -p mod/class
   4. java -m modules mod/class
   5. java -m modules -p mod/class
   6. java -p modules mod/class
   7. java -p modules -m mod/class
2. What is the result of executing the following code snippet?

final int score1 = 8, score2 = 3;

char myScore = 7;

switch (myScore) {

default:

score1:

2: 6: System.out.print("great-");

4: System.out.print("good-"); break;

score2:

1: System.out.print("not good-");

}

* 1. great-good-
  2. good-
  3. not good-
  4. great-good-not-good-
  5. The code does not compile because default is not a keyword in Java.
  6. The code does not compile for a different reason.

1. Which of the following lines can fill in the blank to print true? (Choose all that apply.)

10: public static void main(String[] args) {

11: System.out.println(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_);

12: }

13: private static boolean test(Predicate<Integer> p) {

14: return p.test(5);

15: }

* 1. test(i -> i == 5)
  2. test(i -> {i == 5;})
  3. test((i) -> i == 5)
  4. test((int i) -> i == 5)
  5. test((int i) -> {return i == 5;})
  6. test((i) -> {return i == 5;})

1. Which of the following are valid instance members of a class? (Choose all that apply.)
   1. var var = 3;
   2. Var case = new Var();
   3. void var() {}
   4. int Var() { var \_ = 7; return \_;}
   5. String new = "var";
   6. var var() { return null; }
2. Which of the following types can be inserted into the blank that allows the program to compile successfully? (Choose all that apply.)

1: import java.util.\*;

2: interface CanSwim {}

3: class Amphibian implements CanSwim {}

4: abstract class Tadpole extends Amphibian {}

5: public class FindAllTadPole {

6: public static void main(String[] args) {

7: var tadpoles = new ArrayList<Tadpole>();

8: for (Amphibian amphibian : tadpoles) {

9: \_\_\_\_\_\_\_\_\_\_\_\_ tadpole = amphibian;

10: } } }

* 1. CanSwim
  2. Boolean
  3. Amphibian
  4. Tadpole
  5. Object
  6. None of the above; the program contains a compilation error.

1. Which of the following expressions compile without error? (Choose all that apply.)
   1. int monday = 3 + 2.0;
   2. double tuesday = 5\_6L;
   3. boolean wednesday = 1 > 2 ? !true;
   4. short thursday = (short)Integer.MAX\_VALUE;
   5. long friday = 8.0L;
   6. var saturday = 2\_.0;
   7. None of the above
2. Suppose you have a module named com.vet. Where could you place the following module-info.java file to create a valid module?

public module com.vet {

exports com.vet;

}

* 1. At the same level as the com folder
  2. At the same level as the vet folder
  3. Inside the vet folder
  4. None of the above

1. What is the result of compiling and executing the following program?

1: public class FeedingSchedule {

2: public static void main(String[] args) {

3: var x = 5;

4: var j = 0;

5: OUTER: for (var i = 0; i < 3;)

6: INNER: do {

7: i++;

8: x++;

9: if (x > 10) break INNER;

10: x += 4;

11: j++;

12: } while (j <= 2);

13: System.out.println(x);

14: } }

* 1. 10
  2. 11
  3. 12
  4. 17
  5. The code will not compile because of line 5.
  6. The code will not compile because of line 6.

1. Which statement about the following method is true?

5: public static void main(String... unused) {

6: System.out.print("a");

7: try (StringBuilder reader = new StringBuilder()) {

8: System.out.print("b");

9: throw new IllegalArgumentException();

10: } catch (Exception e || RuntimeException e) {

11: System.out.print("c");

12: throw new FileNotFoundException();

13: } finally {

14: System.out.print("d");

15: } }

* 1. It compiles and prints abc.
  2. It compiles and prints abd.
  3. It compiles and prints abcd.
  4. One line contains a compiler error.
  5. Two lines contain a compiler error.
  6. Three lines contain a compiler error.
  7. It compiles but prints an exception at runtime.

1. Which of the following are true statements? (Choose all that apply.)
   1. The JDK contains a compiler.
   2. The JVM contains a compiler.
   3. The javac command creates a file containing bytecode.
   4. The java command creates a file containing bytecode.
   5. The JDK is contained in the JVM.
   6. The JVM is contained in the JDK.
2. Which lines in Tadpole give a compiler error? (Choose all that apply.)

1: package animal;

2: public class Frog {

3: protected void ribbit() { }

4: void jump() { }

5: }

1: package other;

2: import animal.\*;

3: public class Tadpole extends Frog {

4: public static void main(String[] args) {

5: Tadpole t = new Tadpole();

6: t.ribbit();

7: t.jump();

8: Frog f = new Tadpole();

9: f.ribbit();

10: f.jump();

11: } }

* 1. 5
  2. 6
  3. 7
  4. 8
  5. 9
  6. 10

1. What is the output of the following program?

1: class Deer {

2: public Deer() {System.out.print("Deer");}

3: public Deer(int age) {System.out.print("DeerAge");}

4: protected boolean hasHorns() { return false; }

5: }

6: public class Reindeer extends Deer {

7: public Reindeer(int age) {System.out.print("Reindeer");}

8: public boolean hasHorns() { return true; }

9: public static void main(String[] args) {

10: Deer deer = new Reindeer(5);

11: System.out.println("," + deer.hasHorns());

12: } }

* 1. ReindeerDeer,false
  2. DeerAgeReindeer,true
  3. DeerReindeer,true
  4. DeerReindeer,false
  5. ReindeerDeer,true
  6. DeerAgeReindeer,false
  7. The code will not compile because of line 4.
  8. The code will not compile because of line 12.

1. What is printed by the following code? (Choose all that apply.)

int[] array = {6,9,8};

List<Integer> list = new ArrayList<>();

list.add(array[0]);

list.add(array[2]);

list.set(1, array[1]);

list.remove(0);

System.out.println(list);

System.out.println("C" + Arrays.compare(array,

new int[] {6, 9, 8}));

System.out.println("M" + Arrays.mismatch(array,

new int[] {6, 9, 8}));

* 1. [8]
  2. [9]
  3. [Ljava.lang.String;@160bc7c0
  4. C-1
  5. C0
  6. M-1
  7. M0
  8. The code does not compile.

1. Which statements about the following program are true? (Choose all that apply.)

1: public class Grasshopper {

2: public Grasshopper(String n) {

3: name = n;

4: }

5: public static void main(String[] args) {

6: Grasshopper one = new Grasshopper("g1");

7: Grasshopper two = new Grasshopper("g2");

8: one = two;

9: two = null;

10: one = null;

11: }

12: private String name;

13: }

* 1. Immediately after line 8, no Grasshopper objects are eligible for garbage collection.
  2. Immediately after line 9, no Grasshopper objects are eligible for garbage collection.
  3. Immediately after line 8, only one Grasshopper object is eligible for garbage collection.
  4. Immediately after line 9, only one Grasshopper object is eligible for garbage collection.
  5. Immediately after line 10, only one Grasshopper object is eligible for garbage collection.
  6. The code does not compile.

1. Which of the following statements about error handling in Java are correct? (Choose all that apply.)
   1. Checked exceptions are intended to be thrown by the JVM (and not the programmer).
   2. Checked exceptions are required to be handled or declared.
   3. Errors are intended to be thrown by the JVM (and not the programmer).
   4. Errors are required to be caught or declared.
   5. Runtime exceptions are intended to be thrown by the JVM (and not the programmer).
   6. Runtime exceptions are required to be handled or declared.
2. Which of the following are valid method modifiers that cannot be used together in a method declaration? (Choose all that apply.)
   1. null and final
   2. abstract and private
   3. public and private
   4. nonstatic and abstract
   5. private and final
   6. abstract and static
   7. protected and abstract
3. Which of the following are true to sort the list? (Choose all that apply.)

13: int multiplier = 1;

14: multiplier \*= -1;

15: List<Integer> list = List.of(99, 66, 77, 88);

16: list.sort(\_\_\_\_\_\_\_\_\_\_\_\_\_);

* 1. Line 14 must be removed for any of the following lambdas to compile.
  2. Line 14 may remain for any of the following lambdas to compile.
  3. (x, y) -> multiplier \* y.compareTo(x)
  4. x, y -> multiplier \* y.compareTo(x)
  5. (x, y) -> return multiplier \* y.compareTo(x)
  6. x, y -> return multiplier \* y.compareTo(x)