# Mock Exam 808 OCA

**Question 01.**

Which of the following are valid Java identifiers?

(Choose all that apply)

**A.** A$B

**B.** \_helloWorld

**C.** true

**D.** java.lang

**E.** Public

**F.** 1980\_s

**Question 02.**

Suppose we have a class named Rabbit. Which of the following statements are true?

(Choose all that apply)

1: public class Rabbit {

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

3: Rabbit one = new Rabbit();

4: Rabbit two = new Rabbit();

5: Rabbit three = one;

6: one = null;

7: Rabbit four = one;

8: three = null;

9: two = null;

10: two = new Rabbit();

11: System.gc();

12: } }

**A.** The Rabbit object from line 3 is first eligible for garbage collection immediately following line 6.

**B.** The Rabbit object from line 3 is first eligible for garbage collection immediately following line 8.

**C.** The Rabbit object from line 3 is first eligible for garbage collection immediately following line 12.

**D.** The Rabbit object from line 4 is first eligible for garbage collection immediately following line 9.

**E.** The Rabbit object from line 4 is first eligible for garbage collection immediately following line 11.

**F.** The Rabbit object from line 4 is first eligible for garbage collection immediately following line 12.

**Question 03.**

Which of the following lines of code compile?

(Choose all that apply)

**A.** int i1 = 1\_234;

**B.** double d1 = 1\_234\_.0;

**C.** double d2 = 1\_234.\_0;

**D.** double d3 = 1\_234.0\_;

**E.** double d4 = 1\_234.0;

**F.** None of the above.

**Question 04.**

What data type (or types) will allow the following code snippet to compile?

(Choose all that apply)

byte x = 5;

byte y = 10;

\_\_\_\_\_ z = x + y;

**A.** int

**B.** long

**C.** boolean

**D.** double

**E.** short

**F.** byte

**Question 05.**

What is the output of the following code?

Choose 1 option.

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.

**Question 06.**

What is the result of the following code snippet?

3: final char a = 'A', d = 'D';

4: char grade = 'B';

5: switch(grade) {

6: case a:

7: case 'B': System.out.print("great");

8: case 'C': System.out.print("good"); break;

9: case d:

10: case 'F': System.out.print("not good");

11: }

**A.** great

**B.** greatgood

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

**D.** The code will not compile because of line 6.

**E.** The code will not compile because of lines 6 and 9.

**Question 07.**

What is the output of the following code snippet?

Choose 1 option.

3: do {

4: int y = 1;

5: System.out.print(y++ + " ");

6: } while(y <= 10);

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

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

**C.** 1 2 3 4 5 6 7 8 9 10 11

**D.** The code will not compile because of line 6.

**E.** The code contains an infinite loop and does not terminate.

**Question 08.**

What is the output of the following code snippet?

3: int count = 0;

4: ROW\_LOOP: for(int row = 1; row <=3; row++)

5: for(int col = 1; col <=2 ; col++) {

6: if(row \* col % 2 == 0) continue ROW\_LOOP;

7: count++;

8: }

9: System.out.println(count);

**A.** 1

**B.** 2

**C.** 3

**D.** 4

**E.** 6

**F.** The code will not compile because of line 6.

**Question 09.**

Which of the following are output by this code?

(Choose all that apply)

3: String s = "Hello";

4: String t = new String(s);

5: if ("Hello".equals(s)) System.out.println("one");

6: if (t == s) System.out.println("two");

7: if (t.equals(s)) System.out.println("three");

8: if ("Hello" == s) System.out.println("four");

9: if ("Hello" == t) System.out.println("five");

**A.** one

**B.** two

**C.** three

**D.** four

**E.** five

**F.** The code does not compile.

**Question 10.**

Which are the results of the following code?

(Choose all that apply)

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.

**Question 11.** Which of the following can replace line 4 to print "avaJ"?

(Choose all that apply)

3: StringBuilder puzzle = new StringBuilder("Java");

4: // INSERT CODE HERE

5: System.out.println(puzzle);

**A.** puzzle.reverse();

**B.** puzzle.append("vaJ$").substring(0, 4);

**C.** puzzle.append("vaJ$").delete(0, 3).deleteCharAt(puzzle.length() - 1);

**D.** puzzle.append("vaJ$").delete(0, 3).deleteCharAt(puzzle.length());

**E.** None of the above.

**12.** What is the result of the following statements?

3: ArrayList<Integer> values = new ArrayList<>();

4: values.add(4);

5: values.add(5);

6: values.set(1, 6);

7: values.remove(0);

8: for (Integer v : values) System.out.print(v);

**A.** 4

**B.** 5

**C.** 6

**D.** 46

**E.** 45

**F.** An exception is thrown.

**G.** The code does not compile.

**13.** Which are true statements? (Choose all that apply)

**A.** An immutable object can be modified.

**B.** An immutable object cannot be modified.

**C.** An immutable object can be garbage collected.

**D.** An immutable object cannot be garbage collected.

**E.** String is immutable.

**F.** StringBuffer is immutable.

**G.** StringBuilder is immutable.

**14.** Which of the following can be inserted into the blank to create a date of June 21, 2014? (Choose all that apply)

import java.time.\*;

public class StartOfSummer {

public static void main(String[] args) {

LocalDate date = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

}

}

**A.** new LocalDate(2014, 5, 21);

**B.** new LocalDate(2014, 6, 21);

**C.** LocalDate.of(2014, 5, 21);

**D.** LocalDate.of(2014, 6, 21);

**E.** LocalDate.of(2014, Calendar.JUNE, 21);

**F.** LocalDate.of(2014, Month.JUNE, 21);

**15.** Which of the following can be inserted into the blank to create a date June 21, 2014?

(Choose all that apply)

import java.time.\*;

public class StartOfSummer {

public static void main(String[] args) {

LocalDate date = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

}

}

**A.** new LocalDate(2014, 5, 21);

**B.** new LocalDate(2014, 6, 21);

**C.** LocalDate.of(2014, 5, 21);

**D.** LocalDate.of(2014, 6, 21);

**E.** LocalDate.of(2014, Calendar.JUNE, 21);

**F.** LocalDate.of(2014, Month.JUNE, 21);

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

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.

**17.** 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() { }

**18.** 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]);

**19.** 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

**20.** 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.

**21.** 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>);

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

LocalDate date = LocalDate.of(2018, Month.APRIL, 30);

date.plusDays(2);

date.plusYears(3);

System.out.println(date.getYear() + " " + date.getMonth() + " " + date.getDayOfMonth());

**A.** 2018 APRIL 2

**B.** 2018 APRIL 30

**C.** 2018 MAY 2

**D.** 2021 APRIL 2

**E.** 2021 APRIL 30

**F.** 2021 MAY 2

**G.** A runtime exception is thrown.

**23.** 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: 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

**24.** 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.

**25.** 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() { } }

**26.** 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.** ucrcr

**C.** u ucrcr

**D.** u u curcur

**E.** u u ucrcr

**F.** ur ur urc

**G.** The code does not compile.

**27.** Given:

Class A has a reference to an object of class X

Class B is-a A

A calls public instance methods of class Y

B accesses public instance fields of class Z

Which of the following statements are correct?

**Select 3 option(s)**

a) Class B has high cohesion.

b) Level of cohesion of A and B cannot be determined.

c) Z violates encapsulation.

d) B violates encapsulation.

e) A is loosely coupled to Y

f) B has-a Y

**28.** 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

**29.** 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).

**30.** The options below contain the complete contents of a file (the name of the file is not specified). Which of these options can be run with the following command line once compiled?

java main

**Select 1 option**

**A.** //in file main.java

class main {

public void main(String[] args) { System.out.println("hello"); } }

**B.** //in file main.java

public static void main4(String[] args) { System.out.println("hello"); }

**C.** //in file main.java

public class anotherone{ }

class main {

public static void main(String[] args) { System.out.println("hello"); } }

**D.** //in file main.java

class anothermain{

public static void main(String[] args) { System.out.println("hello2"); } }

class main {

public final static void main(String[] args) { System.out.println("hello"); } }

**31.** Given the following,

1. interface Base {

2. boolean m1 ();

3. byte m2(short s);

4. }

Which code fragments will compile?

(Choose 2 options)

**A.** interface Base2 implements Base { }

**B.** abstract class Class2 extends Base {

public boolean ml() { return true; } }

**C.** abstract class Class2 implements Base { }

**D.** abstract class Class2 implements Base {

public boolean m1() { return (true); } }

**E.** class Class2 implements Base {

boolean m1( ) { return false; }

byte m2(short s) { return 42; } }

**32.** 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()

**33.** 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.

**34.** 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.

**35.** Which of the following statements are true? (Choose all that apply)

**A.** Runtime exceptions are the same thing as checked exceptions.

**B.** Runtime exceptions are the same thing as unchecked exceptions.

**C.** You can declare only checked exceptions.

**D.** You can declare only unchecked exceptions.

**E.** You can handle only Exception subclasses.

**36.** 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"; });

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

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.

**38.** When are you required to use a finally block in a regular try statement (not a try-withresources)?

**A.** Never.

**B.** When the program code doesn’t terminate on its own.

**C.** When there are no catch blocks in a try statement.

**D.** When there is exactly one catch block in a try statement.

**E.** When there are two or more catch blocks in a try statement.

**39.** Which of the following exceptions are thrown by the JVM? (Choose all that apply)

**A.** ArrayIndexOutOfBoundsException

**B.** ExceptionInInitializerError

**C.** java.io.IOException

**D.** NullPointerException

**E.** NumberFormatException

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

3: try {

4: return a / b;

5: } catch (RuntimeException e) {

6: return -1;

7: } catch (ArithmeticException e) {

8: return 0;

9: } finally {

10: System.out.print("done");

11: }

**A.** -1

**B.** 0

**C.** done-1

**D.** done0

**E.** The code does not compile.

**F.** An uncaught exception is thrown.

**41.** Which of the following can be inserted into Lion to make this code compile?

(Choose all that apply)

class HasSoreThroatException extends Exception {}

class TiredException extends RuntimeException {}

interface Roar {

void roar() throws HasSoreThroatException;

}

class Lion implements Roar {

// INSERT CODE HERE

}

**A.** public void roar(){}

**B.** public void roar() throws Exception{}

**C.** public void roar() throws HasSoreThroatException{}

**D.** public void roar() throws IllegalArgumentException{}

**E.** public void roar() throws TiredException{}