1. Given the content of the in. txt file:

0123456789

and the code fragment:

char[] buffer = new char[8];

int count = 0;

try(FileReader in = new FileReader(“in.txt”);

FileWriter out = new FileWriter(“out.txt”)) {

while((count = in.read(buffer)) != -1) {

out.write(buffer);

}

}

What is the content of the out.txt file?

a. 0123456789234567

b. 012345678901234

c. 012345678

d. 01234567801234

e. 01234567

f. 0123456789

2. Given the code fragment:

//line n1

String input = console.readLine(“Input a number: “);

Int number = Integer.parseInt(input);

If(number % 2 == 0){

System.out.println(number + “ is even.”);

} else {

System.out.println(number + “ is odd”);

}

Which code at line n1, obtains the java.io.Console object?

a. Console console = new Console(new InputStreamReader(System.in));

b. Console console = System.console();

c. Console console = System.console(System.in);

d. Console console = Console.getIntance();

e. Console console = new Console(System.in);

3. Given the code fragment:

LocalDate now = LocalDate.now();

LocalDate this\_labourDay = LocalDate.of (2021, Month.MAY, 1);

LocalDate next\_labourDay = this\_labourDay.plusYears (1);

LocalDate add\_week = next\_labourDay.plusDays (7);

Period p = now.until (add\_week);

System. out. println(p);

Assume LocalDate.now() returns 17th Aug 2021.

What is the result?

a. P8M21D

b. 21D8M

c. 08-MAY-2022

d. 1Y0M7D

4. Given the code fragment:

Duration duration = Duration.ofMillis(5000);

System.out.print(duration) ;

duration = Duration.ofSeconds (60);

System.out.print(duration);

Period period = Period.ofDays (6);

System.out. print (period);

What is the result?

a.551M6D

b. PT5SPT1MP6D

c. PT5000SPT60MP6D

d. 5000S60M6D

5. Daylight Saving Time DST) is the practice of advancing clocks at the start of spring by one hour and adjusting them backward by one hour in autumn.

Considering that in 2021, DST in Chicago (Illinois) ended on November 7th at 2 AM, and given the fragment:

ZoneId zoneID = ZoneId.of("America/Chicago");

ZonedDateTime zdt = ZonedDateTime.of(

LocalDate.of (2021, 11, 7),

LocalTime.of (1, 30),

zoneID

);

ZonedDateTime anHourLater = zdt.plusHours(1);

System.out.println(zdt.getHour () == anHourLater.getHour());

System.out.print(zdt.getOffset().equals(anHourLater.getOffset()));

What is the output?

a. false

true

b. true

true

c. false

false

d. true

false

6. Given the course table:

|  |  |  |  |
| --- | --- | --- | --- |
| COURSE\_ID | COURSE\_NME | COURSE\_FEE | COURSE\_LEVEL |
| 1021 | Java Programmer | 600.00 | 1 |
| 1022 | Java Architect | 600.00 | 2 |
| 1023 | Java Master | 600.00 | 2 |

Given the code fragment:

try (Connection con = DriverManager.getConnection(connectionString)) {

con.setAutoCommit(false);

Statement stmt = con.createStatement();

Savepoint sp1 = con.setSavepoint(“p1”);

String qry = “UPDATE course SET course\_fee = 1000 where COURSE\_ID = 1021”;

stmt.execute(qry);

Savepoint sp2 = con.setSavepoint(“p2”);

String qry2 = “UPDATE course SET course\_fee = 800 where COURSE\_ID = 1023”;

Stmt.execute(qry2);

con.rollback(sp1);

} catch(SQLException sqlException) {

System.out.println(sqlException);

}

What is the result?

a. The program executes and only the row with course\_id 1023 is updated.

b. The program throws an exception.

c. The program executes and only the row with course\_id 1021 is updated.

d. The program executes and the rows of the table are not updated.

7. Which statement is true about modules?

a. Only automatic modules are on the module path.

b. Automatic and named modules are on the module path.

c. Only named modules are on the module path.

d. Only unnamed modules are on the module path.

e. Automatic and unnamed modules are on the module path.

8. Which two statements are true about modules in the Java Platform Module System?

a. The module-info.java has to be located in the Java bin directory.

b. They are a group of related Java packages and resources such as XML files, images, and properties file.

c. Packages within the module are hidden by default.

d. The classes in a module are loaded using class-path.

e. They support versioning of Java modules.

9. Given:

interface IFace {

public void m1();

public default void m2() {

System.out.println(“m2”);

}

public static void m3(){

System.out.println(“m3”);

}

private void m4() {

System.out.println(“m4”);

}

}

Class MyC implements IFace {

public void m1() {

System.out.println(“Hello”);

}

}

Which two method invocations execute?

a. IFace.m4();

b. IFace.m3();

c. new MyC().m2();

d. IFace myClassObj = new MyC();

myClassObj.m3();

e. IFace.m2();

f. IFace myClassObj = new MyC();

myClassObj.m4();

10. Given the code fragment:

abstract sealed interface SInt permits Story, Art {

default String getTitle() { return "Book Title" ; }

}

Which set of class definitions compiles?

a. non-sealed interface Story extends SInt {}

class Art implements SInt {}

b. interface Story extends Sin{}

interface Art extends SInt {}

c. sealed interface Story extends SInt {}

non-sealed class Art implements SInt {}

d. non-sealed interface Story extends SInt {}

non-sealed interface Art extends SInt {}

e. public interface Story extends Sint {}

public interface Art extends SInt {}

11. Given the code fragment:

Stream <String> s1 = Stream.of(“A”,”B”,”C”,”B”);

Steam <String> s2 = Stream.of(“A”,”D”,”E”);

Stream.concat(s1,s2).parallel().distinct().forEach(element -> System.out.print(element));

What is the result?

a. ADEABCB // the order of elements is unpredictable

b. ABCDE // the order of elements is unpredictable

c. ABCDE

d. ABBCDE // the order of elements is unpredictable

12. Given:

public class Test {

public static void main(String[] args) {

List<String> elements = Arrays.asList(“car”,”truck”,”car”,”bicycle”,”car”,”truck”,”motorcycle”);

Map<String, Long> outcome = elements.stream().collect(Collectors.groupingBy(Function.identity(),Collectors.counting()));

System.out.println(outcome);

}

}

What is the result?

a. {3:bicycle, 0:car, 6:motorcycle, 5: truck}

b. {bicycle=1, car 3, motorcycle=1, truck=2}

c. {bicycle, car, motorcycle, truck}

d. {bicycle-1, car=1, notoreycle=7, truck=7}

e. Compilation fails.

13. Given the code fragment:

List<String> 1st = new ArrayList<>();

List<String> sList = new CopyOnWriteArrayList<>(1st);

Runnable r = () -> {

for (int i = 0; i<3; i++) { sList.add(“Item “ + i);

}

};

Thread t1 = new Thread(r, “t1”);

t1.start();

Thread t2 = new Thread(r,”t2”);

t2.start();

Which code fragment prints the elements of sList?

a. try {

t2.join();

sList.forEach (n -> {

System.out.println(n);

});

} catch (InterruptedException ex) {

System.out.println(ex);

}

b. synchronized(sList) {

sList.forEach(n -> {

System.out.println(n);

});

}

c. try{

t1.yield();

synchronized (sList){

sList.forEach (n -> {

System.out.println(n);

});

} catch (InterruptedException ex) {

System.out.println(ex);

}

d. sList.forEach(n -> {

System.out.println(n);

});

14. Given:

final class Folder { // line n1

// line n2

public void open(){

System.out print(“Open “);

}

}

public class Test {

public static void main(String[] args) throws Exception {

try (Folder f = new Folder()) {

f.open();

}

}

}

Which two modifications enable the code to print Open Close?

a. At Line n2, insert:

final void close(){

System.out.print(“Close “);

}

b. Replace line n1 with:

class Folder extends Exception {

c. At line n2, insert:

public void close() throws IOException {

System.out.print ("Close ");

}

d. Replace line n1 with:

class Folder extends Closeable {

e. Replace line n1 with:

class Folder implements AutoCloseable {

15. Given:

package p1;

class ZeroValueException extends Exception{}

and the code fragment:

int a[] = {0,2,4);

try{

try{

if (a[0] == 0) throw new ZeroValueException();

} catch (ZeroValueException | NullPointerException zve) {

throw zve;

}

} catch (Exception e) {

System.out.printIn("Caught " + e);

}

What is the result?

a. Caught p1.ZeroValueException

b. Caught null

c. Caught java.lang.Exception

d. Caught java.lang.NullPointerException

16. Given the code fragments:

class Test {

volatile int x = 1;

AtomicInteger xObj = new AtomicInteger(1);

and

public static void main (String[] args) {

Test t = new Test ();

Runnable r1 = () -> {

Thread trd = Thread.currentThread();

while (t.x < 3) {

System.out.print(trd.getName()+" : "+t.x+" : ");

t.x++;

}

};

Runnable t2 = () → {

Thread trd = Thread.currentThread();

while (t.x0bj.get () < 3) {

System.out.print(trd.getName()+” : “+ t.x0bj.get ()+" : ");

t.x0bj.getAndIncrement();

}

};

Thread t1 = new Thread (r1, "t1");

Thread t2 = new Thread (r2, "t2");

t1.start();

t2.start();

}

Which is true?

a. The program prints t1 : 1 : t2 : 1 : t1 : 2 : t2. : 2 : in random order.

b. The program prints t1: 1 : t2: 1 : t1 : 2 : t2 : 2 :

c. The program prints an exception.

d. The program prints t1: 1 : t2: 1 : t1 : 1 : t2 : 1 : indefinitely.

17. Which statement is true?

a. IllegalStateException is thrown if a thread in waiting state is moved back to runnable.

b. A thread in waiting state must handle InterruptedException.

c. After the timed wait expires, the waited thread moves to the terminated state.

d. A thread in waiting state consumes CPU cycles.

18. Given the code fragment:

ExecutorService executorService = Executors.newSingleThreadExecutor();

Set<Callable<String>> workers = new HashSet<Callable<String>>();

workers.add(new Callable<String> () {

public String call() throws Exception {

return "1";

}

});

workers.add(new Callable<String> () {

public String call() throws Exception {

return "2";

}

});

workers.add(new Callable<String>() {

public String call() throws Exception {

return "3";

}

});

Which code fragment invokes all callable objects in the workers set?

a. List<Future<String>> futures = executorService.invokeAny(workers);

for(Future<String> future : futures) {

System.out.printIn(future.get ());

}

b. for (int i=0; i<3;i++) {

String result = executorService.invokeAny (cThreads);

System.out.println(result);

}

c. executorService.submit(cThreads);

d. List<Future<String>> futures = executorService.invokeAll(workers);

for (Future<String> future : futures) {

System.out.printin(future.get());

}

19. Given the code fragment:

Path p1 = Paths.get ("fldr1\\fldr2\\file1.txt");

Path p2 = Paths.get ("fldr3\\file1.txt");

System.out.println(p1.resolve(p2));

System.out.println(p1.relativize(p2));

What is the result?

a. fldr1\fldr2\file1.txt\fldr3\file1.txt

fldr3\filel.txt

b. fldr1\fldr2\fldr3\file1.txt

..\..\..\..\fldr3\file1.txt

c. fldr1\fldr2\file1.txt\fldr3

..\..\..\fldr3\

20. Given the definition of Doc class:

package p1;

public abstract sealed class Doc permits WordDoc {

String docName;

public abstract void printDoc();

public String getNare() { return docName; }

Which statement is true?

a. The WordDoc class should exist within p1 of module1.

b. The getName() method should be abstract.

c. The WordDoc class could exist within another package, p2, of module1.

d. The Doc dass must declare non-abstract.

21. Given:

class A { public void mA() {System.out.println("mA"); }}

class B extends A {public void mB() {System.out.println("mB");}}

class C extends B {public void mC() {System.out.println("mC");}}

public class App {

public static void main (String[] args) {

A bobj = new B();

A cobj = new C();

if (cobj instanceof B v) {

v.mB ();

if (v instanceof C v1) { v1.mC(); }

} else {

cobj.mA();

}

}

}

What is the result?

a. mB

mC

b. mB

c. mA

d. mB

mA

22. Given the code fragment:

var list = new CopyOnWriteArraylist<>(List.of ("Java", "Duke", "Cafe", "Babe"));

Thread t1 = new Thread (() -> {

try{

Thread.sleep (150);

} catch (InterruptedException e) {

}

int i = 0;

for (var s : list) {

list.set(i++, s.toUpperCase()):

}

System.out.print(list + “ “);

});

Thread t2 = new Thread (() -> {

for (var s : list) {

try {

Thread.sleep (50);

} catch (InterruptedException e) {

}

System.out.print (s +” “);

}

});

t1.start();

t2.start();

What is the expected result?

a. JAVA DUKE [JAVA, DUKE, CAFE, BABE] CAFE BABE is printed.

b. A ConcurrentModificationException is thrown.

c. An IndexOutOfBoundsException is thrown.

d. Java Duke [JAVA, DUKE, CAFE, BABE] Cafe Babe is printed and order of elements is unpredictable.

23. Given the code fragment:

Stream.of(3,6,9,12, 15,18,21,24,27)

.takeWhile(s -> s % 3 == 0)

.dropWhile(s -> s % 2 == 0)

.limit(3).forEach(s -> System.out.print (s + " "));

What is the result?

a. 3 6 9

b. 3 9 15 21

c. 15 21 27

d. 3 9 15

24. Given the code fragment:

List<Integer> l1 = List.of(1, 3, 5, 7, 9);

List<Integer> l2 = List.of (2, 4, 6, 8);

List<List<Integer>> l3 = List.of (11, 12);

//line n1

System.out.println(l4);

Which code fragment at line n1 enables l4 to contain numbers greater than 5 from the lists l1 and l2?

a. List<Integer> l4 = l3.stream()

.flatMap(x -> x.stream().filter(y -> y > 5).toList();

b. List<Integer> l4 = l3.stream()

.map(x -> x.stream()).filter(y -> y > 5)

.collect(Collectors.toList());

c. List<Integer> l4 = l3.stream()

.flatMap(x -> x.stream().filter(y -> y > 5)

.collect(Collectors.toList());

d. List<Integer> l4 = l3.stream()

.filter(y -> y > 5).collect();

25. Given the code fragment:

List<String> specialDays = List.of ("NewYear", "Valentines", "Spring", "Labour");

System.out.print(specialDays.stream().allMatch(s ->s. equals ("Labour")));

System.out.print(“ “+ specialDays.stream().anyMatch (s -> s.equals("Labour")));

System.out.print(“ “ + specialDays.stream().noneMatch (s -> s. equals ("Halloween")));

System.out.print(“ “ + specialDays.stream().findFirst());

What is the result?

a. false true true Optional [NewYear]

b. true true false NewYear

c. 0 1 0 Optional [NewYear]

d. 0 1 1 0

26. Which statement is true about migration?

a. Unnamed modules are automatic modules in a top-down migration.

b. Every module is moved to the module path in a bottom-up migration.

c. Every module is moved to the module path in a top-down migration.

d. The required modules migrate before the modules that depend on them in a top-down migration.

27. Assume you have an automatic module from the module path display-ascii-0.2.jar.

Which name is given to the automatic module based on the given JAR file?

a. display-ascii-0.2

b. display-ascii

c. display-ascii-0

d. display.ascii

28. Given the code fragment:

List<String> 1st = new ArrayList<String>();

1st.add("e1");

1st.add(“e2");

1st.add("e3");

Deque<String> dq1 = new ArrayDeque<>(1st);

System.out.printIn(dql.offer("e4"));

System.out.println(dq1.pollFirst());

System.out.println(dq1.peekFirst());

System.out.println(dq1.pop());

System.out.println(dq1);

What is the result?

a. true

e1

e2

e2

[e3, e4]

b. true

e1

e1

e2

[e3, e4]

c. 1

e1

e1

e1

[e1, e3]

a. true

e1

e1

e2

29. Given the code fragment:

class Book {

String author;

String title;

Book(String authorName, String title) {

this.author = authorName;

this.title = title;

}

}

class SortBook {

public static void main(String[] args) {

List<Book> books = List.of(new Book(“A1”,”T1”), new Book(“A1”,”T2”)); // Line n1

books.sort(Book a, Book b) -> a.title.compareTo(b.title));

System.out.println(books);

}

}

Which action sorts the book list?

a. At Line n1, convert books type to mutable array type.

b. At Line n2, replace books.sort() with books.stream().sort().

c. At Line n1, convert books type to mutable ArrayList type.

d. At Line n2, replace compareTo() with compare().

30. Given the code fragment:

int a = 2;

int b = ~a;

int c = a^b;

boolean d = a < b & a > c++;

System.out.println(d + " “ + c);

boolean e = a > b && a > c++;

System.out.println(e + " “ + c);

What is the result?

a. false 1

false 2

b. false 1

true 2

c. true 1

false 2

d. false 0

true 2

31. Given the code fragment:

float x = Math.round((1.0/3) \*100) /10f/(((int) 'a' == 0Xa) ? 0 : 3);

What is the result?

a. 0.0

b. Compilation fails.

c. 1

d. 1.1

e. 0

f. An ArithmeticException is thrown.

32. Given the Product class:

import java.io.\*;

public class Product implements Serializable {

private static float averagePrice = 2.99f;

private String description;

private transient float price;

public Product(String description, float price) {

this.description = description;

this price = price;

}

public void readobject(ObjectInputStream in)

throws I0Exception, ClassNotFoundException {

in.defaultReadObject();

price = averagePrice;

}

public String toString() {

return description + “ “ + price +" “ + averagePrice;

and the Shop class:

import java.io.\*;

public class Shop {

public static void main (String[] args) {

Product p = new Product("Cookie", 3.99f);

try {

try (ObjectOutputStream out = new ObjectOutputStream(new FileOutputStream("p.ser"))) {

out.writeObject(p);

}

try (ObjectInputStream in = new ObjectInputStream(new FileInputStream("p.ser"))) {

p = (Product) in.readObject();

}

} catch (Exception e) { e.printStackTrace (); }

System.out.println(p);

}

}

What is the result?

a. An exception is produced at runtime.

b. Compilation fails.

c. Cookie 0.0 2.99

d. Cookie 2.99 2.99

e. Cookie 3.99 2.99

f. Cookie 0.0 0.0

33. Given the code fragment:

class Car implements Serializable {

private static long serialVersionUID = 454L;

String name;

public Car(String name) { this.name = name; }

}

class LuxuryCar extends Car { // line n1

int flag\_HHC;

public LuxuryCar(String name, int flag\_HHC) {

super(name);

this.flag\_HHC = flag\_HHC;

}

public String toString() {

return name + " : " + flag\_HHC;

}

}

and

public static void main(String[] args) { // line n2

Car b = new LuxuryCar("Wagon", 200);

try (ObjectOutputStream oos = new ObjectOutputStream(new FileOutputStream("car.ser"));

ObjectInputStream ois = new ObjectInputStream(new FileInputStream("car.ser"));) {

oos.writeObject(b);

System.out.println((Car)(ois.readObject());

}

} // line n3

Which action prints Wagon : 200?

\* At line n3, replace readObject() with readLine().

\* At line n2, in the main method signature, add throws IOException, ClassCastException.

\* At line n1, implement the java.io.Serializable interface.

\* At line n1, implement the java.io.AutoCloseable interface.

\* At line n2, in the main method signature, add throws IOException, ClassNotFoundException.

\* At line n3, replace Car with LuxuryCar.

34. Given:

import java.io.Serializable;

public class Software implements Serializable {

private String title;

public Software(String title) {

this.title = title;

System.out.print("Software ");

}

public String toString() { return title; }

}

public class Game extends Software {

private int players;

public Game(String title, int players) {

super(title);

this.players = players;

System.out.print("Game ");

}

public String toString() { return super.toString() + " " + players; }

}

import java.io.\*;

public class AppStore {

public static void main(String[] args) {

Software s = new Game("Chess", 2);

try(ObjectOutputStream out = new ObjectOutputStream(new FileOutputStream("game.ser"))) {

out.writeObject(s);

} catch (Exception e) {

System.out.println("write error");

}

try(ObjectInputStream in = new ObjectInputStream(new FileInputStream("game.ser"))) {

s = (Software)in.readObject();

} catch (Exception e) {

System.out.println("read error");

}

System.out.println(s);

}

}

What is the result?

A) Software Game Chess 0

B) Software Game Software Game Chess 2

C) Software Game Chess 2

D) Software Game write error

E) Software Game Software Game Chess 0

F) Software Game read error

35. Which two record definitions are valid?

a.

record Item(int id, String name) {

public Item {

if (name.length() < 3) {

throw new IllegalArgumentException();

}

}

static String qualityCode;

}

b.

record Item(int id, String name) {

public static Item(String n) {

this.name = n;

if (name.length() < 3) {

throw new IllegalArgumentException();

}

}

}

c.

record Item(int id, String name) {

public Item {

if (name.length() < 3) {

throw new IllegalArgumentException();

}

}

static String qualityCode;

}

d.

record Item(int id, String name) {

public static Item {

if (name.length() < 3) {

throw new IllegalArgumentException();

}

}

}

e.

record Item(int id, String name) {

String name = "ho";

public void Item() {

if (name.length() < 3) {

throw new IllegalArgumentException();

}

}

}

f.

record Item(int id, String name) {

static {

if (name.length() < 3) {

throw new IllegalArgumentException();

}

}

String qualityCode;

}

36. Given:

public class App {

public static void main(String[] args) {

Doc ts = new Doc("Sales");

ts.printDoc();

Doc ts1 = new Doc("Purchase");

ts1.printDoc();

}

}

class Doc {

public static Integer dId = 100;

String name;

Doc(String n) {

this.dId = ++dId;

this.name = n;

}

public void printDoc() {

System.out.println(name + " - " + dId + " is printed.");

}

}

What is the result?

a. Sales - 102 is printed. Purchase - 102 is printed.

b. Sales - 101 is printed. Purchase - 102 is printed.

c. Sales - 102 is printed. Purchase - 103 is printed.

d. Sales - 101 is printed. Purchase - 101 is printed.

37. Given the code fragment:

3. public class TestApp {

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

5. int x = 10;

6. if (x > 5) {

7. int grade = 1;

8. } else if (x > 8) {

9. grade = 2;

10. }

11. Predicate < Integer > pred = g -> g == 2;

12. System.out.println("Is it grade 2?" + pred.test(grade));

13. }

14. }

Which action enables the code to compile ?

a. Replace line 11 with :

Predicate <Integer> pred = grade - > grade == 2;

b. At line 12, replace pred.test(grade) with pred.test(g).

c. Declare g at line 5.

d. Declare grade at line 5 instead of line 7.

38. Which two code fragments compile?

a.

class L2 {

public void m(int x) {

var x = 10;

}

}

b.

class L6 {

public static void main(String[] args) {

var x = new ArrayList<>();

x.add(10);

x.add("30");

System.out.println(x);

}

}

c.

class L5 {

public void m() {

var strVar = null;

}

}

d.

class L3 {

public static void main(String[] args) {

var a = 10;

a = "30";

}

}

e.

class A {}

class B extends A {}

class L4 {

public static void main(String[] args) {

var x = new A();

x = new B();

}

}

f.

class L1 {

var x = 100;

}

39. Given the code fragment:

int saleAmount = 10000;

NumberFormat format1 = NumberFormat.getCurrencyInstance(Locale.US);

NumberFormat format2 = NumberFormat.getCompactNumberInstance(Locale.US, Style.SHORT);

String res = format1.format(saleAmount);

System.out.println(res);

format2.format(Integer.valueOf(res));

System.out.println(saleAmount);

What is the result?

a. $10,000.00

$10K

b. $10000.00

$10.00 K

c. $10,000.00

$10,000.00 K

d. $10,000.00

An Exception is thrown.

40. Given the code fragment:

String a = "Hello! Java";

System.out.print(a.indexOf("Java"));

a.replace("Hello!", "Welcome!");

System.out.print(a.indexOf("Java"));

StringBuilder b = new StringBuilder(a);

System.out.print(b.indexOf("Java"));

What is the result?

1. 81111
2. 7107
3. 71010
4. 777
5. 888
6. 8109

41. Given this output:

Hello

Java 17.0.0

Which code fragment leads to its display?

Option 1:

String blk3 = """

String msg = "Hello\n";

String ver = "Java 17.0.0";

System.out.println(msg+ver);

""";

System.out.println(blk3);

Option2:

String s1 = """

Hello""";

String s2 = """

Java 17.0.0""";

String blk4 = s1+s2;

System.out.println(blk4);

Option3:

String blk2 = """

Hello &chr(13)

Java 17.0.0

""";

System.out.println(blk2);

Option4:

String blk1 = """Hello\nJava 17.0.0"""

System.out.println(blk1);

Option5:

String str = "Hello";

String blk5 = str +

"""

Java 17.0.0""";

System.out.println(blk5);

42. Given:

public class App{

String name;

public App(String name){

this.name = name;

}

public static void main(String args[]){

App t1 = new App("t1");

App t2 = new App("t2");

t1 = t2;

t1 = null;

System.out.println("GC");

}

}

Which statement is true while the program prints GC?

a. Only one of the objects previously referenced by t1 is eligible for garbage collection.

b. None of the objects are eligible for garbage collection.

c. Only the object referenced by t2 is eligible for garbage collection.

d. Both the objects previously referenced by t1 are eligible for garbage collection.

43. Given:

class OClass {

private int var1 = 100;

public int var2 = 200;

class IClass {

int var3 = 300;

public int var4 = 400;

int m2() {

return var2;

}

}

public int m1() {

return var1;

}

}

and the code fragment:

public class App {

public static void main(String[] args) {

OClass.IClass inner = new OClass().new IClass();

}

}

Which three members can be accessed by the inner reference variable?

[ ] var4

[ ] var3

[ ] m2

[ ] m1

[ ] var2

[ ] var1

44. Given:

public class Main {

void print(int i) {

System.out.println("hello");

}

void print(long j) {

System.out.println("there");

}

public static void main(String[] args) {

new Main().print(0b1101\_1010);

}

}

What is the result?

O there

O A NumberFormatException is thrown.

O hello

O Compilation fails.

45. Given:

public class Test {

public void sum(int a, int b) {

System.out.print("A");

}

public void sum(int a, float b) {

System.out.print("B");

}

public void sum(float a, float b) {

System.out.print("C");

}

public void sum(double... a) {

System.out.print("D");

}

public static void main(String[] args) {

Test t = new Test();

t.sum(10, 15.25);

t.sum(10, 24);

t.sum(10.25, 10.25);

}

}

What is the result?

a. DAD

b. DDD

c. BAC

d. BAD

46. Given:

public class Test {

static interface Animal {

}

static class Dog implements Animal {

}

private static void play(Animal a) {

System.out.print("flips");

}

private static void play(Dog d) {

System.out.print("runs");

}

public static void main(String[] args) {

Animal a1 = new Dog();

Dog a2 = new Dog();

play(a1);

play(a2);

}

}

What is the result?

O flipsruns

O flipsflips

O runsruns

O runsflips

O Compilation fails.

47. Given:

public enum Designation {

CEO('A'), CMO('B'), CTO('C'), CFO('D');

char c;

private Designation (char c) {

this.c = c;

}

}

and the code fragment:

System.out.println(Designation.valueOf("CMO"));

System.out.println(Designation.values()[0].name());

What is the result?

a. B

A

b. CMO

A

c. B

CEO

d. CMO

CEO

48. Given:

public enum Design {

CEO('A'), CMO('B'), CTO('C'), CFO('D');

char c;

private Design (char c) {

this.c = c;

}

}

and the code fragment:

Arrays.stream(Design.values()).dropWhile(s -> s.equals(Design.CMO));

switch (Design.valueOf("CMO")) {

case CEO -> System.out.println("Executive");

case CMO -> System.out.println("Marketing");

case CFO -> System.out.println("Finance");

case CTO -> System.out.println("Technical");

default -> System.out.println("Undefined");

}

What is the result?

a. Marketing

Undefined

b. Marketing

Finance

Technical

c. Marketing

d. Undefined

49. Given the code fragment:

Integer rank = 4;

switch (rank) {

case 1,4 -> System.out.println("Range1");

case 5,8 -> System.out.println("Range2");

case 9,10 -> System.out.println("Range3");

default -> System.out.println("Not a valid rank.");

}

What is the result?

a. Range1

Range2

Range3

Range1

Not a valid rank.

b. Range1

Not a valid rank.

c. Range1

d. Range1

Range2

Range3

50. Given the code fragment:

Integer grade = 5;

switch (grade) {

case 1,2,3 -> System.out.println("Region 1");

case 4,6 -> System.out.println("Region 2");

case 7,9 -> System.out.println("Region 3");

default -> System.out.println("Invalid grade.");

}

What is the result?

a. Invalid grade.

b. Region 2

Region 3

Invalid grade.

c. Region 2

Region 3

d. Region 2