1.-

Given:

class A{

private static int c=2;

static void m(){

System.out.println(c); //1

}

static class B{

void print(){

c++; //2

m(); //3

}

}

}

And then:

A.B obj=new A.B();

obj.print(); //4

Which is true?

1. It prints 3
2. It prints 2
3. Compilation error at line 1
4. Compilation error at line 2
5. Compilation error at line 3
6. Compilation error at line 4

2.-

What is the result of executing the following code?

**public** **class** Test {

**public** **static** **void** main(String[] args) {

**final** **int** VAL=10;

**int** a=1;

**int** b=VAL;

**for**(;a!=b&&a++<VAL/2;) {

System.***out***.print(a++ +" "+b++ +" ");

}

}

}

1. It prints 2 4 4 6 6 8
2. It prints 2 10 4 11
3. It prints 3 11 5 12
4. It prints 3 10 5 10 6
5. It prints nothing
6. Infinite loop

3.-

Given:

**enum** Data{

***NORMAL***,***MEDIUM***,***SUPER***,***EXTRA***;

}

**public** **class** Test {

**public** **static** **void** main(String[] args) {

Data data=Data.***EXTRA***;

System.***out***.println(

**switch**(data) {

**case** ***NORMAL***->"Must improve";

**case** ***MEDIUM***,***EXTRA***->"Great";

**case** ***SUPER***->"Excellent";

default->"None"

}

);

}

}

Which is the result?

1. it prints "Great"
2. it prints "GreatExcellent"
3. It prints "None"
4. Compilation error because data must be final

4.-

What will be printed when the following code was executed?

StringBuilder sb1=**new** StringBuilder("MyStr");

String s1=sb1.toString().intern();

String s2="""

MyStr""";

StringBuilder sb2=**new** StringBuilder(s2);

System.***out***.print(s1==s2);

System.***out***.print(s2.equals(sb1)); //l1

System.***out***.println(s1.equals(sb2));

1. truetruetrue
2. falsefalsefalse
3. truetruefalse
4. truefalsefalse
5. falsefalsetrue
6. Compilation error at l1

5.-

Given:

**sealed** **class** C1 **permits** C2{

**public** **void** print() {

System.***out***.println("C1");

}

}

**non-sealed** **class** C2 **extends** C1{

**public** **void** print() {

System.***out***.println("C2");

}

}

**class** C3 **extends** C2{

**public** **void** print() { //line1

System.***out***.println("C3");

}

}

**public** **class** TestSealed {

**public** **static** **void** main(String[] args) {

C1 c=(C2)**new** C3();//line2

c.print();

}

}

What will be printed?

1. C1
2. C2
3. C3
4. compilation fails at line1
5. compilation fails at line2

6.-

Given:

List<String> lst=**new** ArrayList<>();

Set<String> st=**new** HashSet<>();

String[] ar= {"car","table","jam","car","jam"};

**for**(String s:ar) {

lst.add(s);

st.add(s);

}

Set<String> st2=**new** HashSet<String>(lst);

List<String> lst2=List.*of*(st.toArray(**new** String[0]));

System.***out***.print(lst.size()+","+st.size()+" ");

System.***out***.print(lst2.size()+","+st2.size()+" ");

What is the result?

1. 5 5 5 5
2. 5 3 3 5
3. 5 3 3 3
4. 5 5 3 3

7.-

Given:

**class** C1{}

**class** C2 **extends** C1{}

**class** C3 **extends** C2{}

And then:

**public** **class** Test {

**public** **static** **void** main(String[] args) {

C1 c=**new** C3();

**if**(c **instanceof** C2 a) {

System.***out***.print("C2");

**if**(a **instanceof** C3 b) {

System.***out***.print("C3");

**if**(b **instanceof** C1 x) {

System.***out***.print("C1");

}

}

}

}

}

What will be printed?

1. C2
2. C2C3
3. C2C3C1
4. C3C3C3
5. It prints noting

8.-

Given:

**class** Main{

**public** **void** print(Collection col) {

System.***out***.println("Main collection");

}

}

**class** Child **extends** Main{

**public** **void** print(Collection col) {

System.***out***.println("Child collection");

}

**public** **void** print(List lst) {

System.***out***.println("Child list");

}

}

And

Main m=**new** Main();

Child c=**new** Child();

Main seg=**new** Child();

List<Integer> lst=**new** ArrayList<>();

Which three are correct? (choose 3)

1. m.print(lst) prints "Main collection"
2. c.print(lst) prints "Child collection"
3. c.print(lst) prints "Child list"
4. seg.print(lst) prints "Main collection"
5. seg.print(lst) prints "Child collection"
6. seg.print(lst) prints "Child list"

9.-

Given:

**public** **record** Order(String product, **int** units,**double** price) {

**//line 1**

}

Which of the following are valid in line1? (choose 2)

1. private int discount=10;
2. public Order(int discount){price=price\*discount/100;}
3. public Order{units+=10;}
4. private static int count=1;
5. public Order(){this(1);}

10.-

Which of the following are RuntimeExceptions? (choose 3)

1. NullPointerException
2. IllegalArgumentException
3. FileNotFoundException
4. OutOfMemoryError
5. NumberFormatException
6. SQLException

11.-

What is true about sealed class? (choose two)

1. sealed class and permits classes must be in the same .java file
2. non-sealed class can´t extends another class
3. sealed class can be abstract
4. sealed class can extends a non-sealed class
5. sealed class can be final

12.-

Which of the following declarations are correct? (choose 3)

1. int a=0b111010;
2. double b=34.7\_4;
3. boolean c=(boolean)1;
4. int d=0489;
5. float e=7.8;
6. double f= Double.valueOf("20");

13.-

Given:

**class** MyException **extends** Exception{}

**public** **class** TestException {

**public** **static** **void** main(String[] args) {

**try** {

System.***out***.print("t");

*met*();

}

**catch**(MyException ex) {

System.***out***.print("r");

**return**;

}

**finally** {

System.***out***.print("f");

}

System.***out***.print("h");

}

**private** **static** **void** met() **throws** MyException {

System.***out***.print(1/0);

**throw** **new** MyException();

}

}

What will be printed?

1. tf and an ArithmeticException is thrown
2. trf
3. trfh
4. tfh
5. tf

14.-

Given:

interface I1{

}

class A implements I1{

}

class B extends A{

}

Which of the following are valid? (choose 2)

1. List<? extends I1> ls=new ArrayList<B>();
2. List<> ls=new ArrayList<I1>();
3. Collection<A> col=new HashSet<B>();
4. Collection<T> col=new ArrayList<?>();
5. Collection<I1> col=new HashSet<>();

15.-

Given:

**enum** Position{

***LEFT***(-1),***RIGTH***(0),***UP***(1),***DOWN***(2);

**int** value;

Position(**int** value){

**this**.value=value;

}

}

**public** **class** Result {

**public** **static** **void** main(String[] args) {

Position p1=Position.*values*()[Integer.*parseInt*(args[0])];

**int** res=**switch**(p1) {

**case** ***LEFT***,***RIGTH***->p1.value;

**case** ***UP***,***DOWN***->Integer.*parseInt*(args[1]);

};

System.***out***.println(Position.*values*()[res].name());

}

}

What will be printed if the Result class is executed with the following command?:

>java Result 1 5

1. 0
2. -1
3. UP
4. RIGTH
5. LEFT
6. An ArrayIndexOutOfBoundsException is thrown

16.-

What will be printed when the following code is executed?

**sealed** **interface** I1 **permits** R1{

**void** print();

}

**record** R1(**int** x, **int** y) **implements** I1{ //line1

R1{

x++;

y++;

}

**public** **void** print() { //line2

System.***out***.println(x+","+y);

}

**public** **void** plus() {

System.***out***.println(x+y);

}

}

**public** **class** Test {

**public** **static** **void** main(String[] args) {

I1 i=**new** R1(10,10);

**if**(i **instanceof** R1 r) { //line3

r.plus();

}**else** {

i.print();

}

}

}

1. 22
2. 11,11
3. Compilation fails at line1
4. Compilation fails at line2
5. Compilation fails at line3

17.-

**public** **class** Test {

**public** **static** **void** main(String[] args) {

**char**[][] cars= {{'a','b','c'},{'1','2','3','4'},{'-','+','/'}};

**int** j=0;

ext:**for**(**int** i=0;i<cars.length;i++) {

**while**(j<3) {

System.***out***.print(cars[i][j]);

**if**(j==2) {

**continue** ext;

}

j++;

}

}

}

}

What will be printed?

1. a1-
2. ab34/
3. abc3/
4. it will throw a ArrayIndexOutOfBoundsException

18.-

Given:

**package com.operations;**

**public interface Actions{**

**void move(int d);**

**}**

And

**package com.operations.implementations;**

**import com.operations.Actions**

**public class ActionsImpl implements Actions{**

**@Override**

**public void move(int d){**

**System.out.println("moving: "+d);**

**}**

**}**

Which two should the module-info file include for it to represent the service provider interface? (choose 2)

1. exports com.operations;
2. provides com.operations.ActionsImpl
3. exports com.operations.ActionsImpl;
4. provides com.operations.Actions with com.operations.implementations.ActionsImpl;
5. requires com.operations;

19.-

Given:

**public** **class** Calc {

**public** **void** print(**int** x, **int** y) {

System.***out***.print(" A");

}

**public** **void** print(**float** x, **float** y) {

System.***out***.print(" B");

}

**public** **void** print(**int** x, **float** y) {

System.***out***.print(" C");

}

**public** **void** print(**int** ...x) {

System.***out***.print(" D");

}

**public** **void** print(**double** ...y) {

System.***out***.print(" E");

}

**public** **static** **void** main(String[] args) {

Calc c=**new** Calc();

c.print(2,4.5);

c.print(0.0,1.1);

c.print(6,4);

}

}

What is the result?

1. C E D
2. C B D
3. E E A
4. E C D
5. E C A

20.-

Given:

**class** OpenClass **implements** Closeable{

//line 1

}

**public** **class** Main {

**public** **static** **void** main(String[] args) {//line 2

**try** (**var** c=**new** OpenClass()){ //line 3

//using c object

}

**catch**(Exception e) {

e.printStackTrace();

}

}

}

What two actions, independently of each other, must be done for the above code to compile? (choose 2)

1. Insert *public void close() thorws IOException{}* in line 1
2. Add *throws Exception* in the declaration main() method in line 2
3. Replace *var c* with *OpenClass c* in line 3
4. Insert *public void close() thorws Exception{}* in line 1
5. Insert public void close() in line 1

21.-

Given:

**interface** I1{

**default** **void** printmethod() {

System.***out***.println("I1");

}

}

**interface** I2{

**default** **void** printmethod() {

System.***out***.println("I2");

}

}

**public** **class** MyClass **implements** I1, I2{

**//line 1**

}

Which of the following methods definitions must be placed on line1 so that the class MyClass compiles correctly and the method prints "I1" when executed?

1. public void printmethod(){I1.super.printmethod();}
2. public void printmethod(){I1.printmethod();}
3. public static void printmethod(){super.I1.printmethod();}
4. public void printmethod(int s){I1.this.printmethod();}

22.-

Given:

**record** City(**int** code, String name) {

**static** **int** *alternative*=10; //l1

**public** City() {//l2

code=*alternative*;//l3

}

}

**public** **class** TestRecord {

**public** **static** **void** main(String[] args) {

**var** city=**new** City(2500,"MyCity");//l4

}

}

Which action enables the code to compile?

Which three are correct? (choose 3)

1. removes the keyword *static* in l1
2. replace City() with City(int code, String name) in l2
3. replace code=alternative; with this(alternative,"other"); in l3
4. replace the keyword var with City in l4

23.-

What is true about application migration?:

1. In a top-dwon migration, lowest-level proyect is migrated first
2. In a bottom-up migration, automatic modules are migrated first
3. In a top-down migration, all proyects are moved to the module-path
4. In a bottom-up migration, lowest-level proyects stay as unnamed modules in the class-path

24.-

Given:

module1:

pck1\

Exam.java

pck2\

module2:

pck3\

And the following definition of Exam.java:

public sealed class Exam permits Result{

...

}

Which of the following two definitions of Result.java is valid?(choose 2)

1. package pck1;

public class Result extends Exam{}

1. package pck2;

public final class Result extends Exam{}

1. package pck3;

public non-sealed class Result extends Exam{}

1. package pck2;

public sealed class Result extends Exam{}

1. package pck1;

public final class Result extends Exam{}

25.-

Which of the following are functional interfaces? (choose 2)

1. interface I1 extends Callable<Integer>{

default Integer call(){return 10;}

}

1. interface I2 extends Serializable{

int car();

}

1. interface I3{

boolean equals();

}

1. interface I4{

String toString();

}

1. interface I5{

static void launch();

}

1. interface I6 extends Runnable{

void run(String a);

}