Vendor: Oracle

Exam Code: 1Z0-808

Exam Name: Java SE 8 Programmer I

Question 131—Question 140

Click to Download All 1Z0-808 Q&As From Lead2pass

QUESTION 131
Given:

```java
public class Test {
public static void main(String[] args) {
try {
String[] arr =new String[4];
arr[1] = "Unix";
arr[2] = "Linux";
arr[3] = "Solarios";
for (String var : arr) {
System.out.print(var + " ");
}
} catch(Exception e) {
System.out.print (e.getClass());
}
}
}
```

What is the result?

A. Unix Linux Solaris

B. Null Unix Linux Solaris

C. Class java.lang.Exception

D. Class java.lang.NullPointerException

Answer: B
Explanation:
null Unix Linux Solarios
The first element, arr[0], has not been defined.

QUESTION 132
Given:

```
public class Series {
    private boolean flag;

    public void displaySeries() {
        int num = 2;
        while (flag) {
            if (num % 7 == 0)
                flag = false;
            System.out.print(num);
            num += 2;
        }
    }
    public static void main(String[] args) {
        new Series().displaySeries();
    }
}
```

What is the result?

A. 2 4 6 8 10 12
B. 2 4 6 8 10 12 14
C. Compilation fails
D. The program prints multiple of 2 infinite times
E. The program prints nothing

Answer: E

QUESTION 133
Which of the following can fill in the blank in this code to make it compile?

```
interface CanFly{

        String type = "A";

        void fly();


_____ String getType(){

            return type;

        }

}
```

A. abstract
B. public
C. default
D. It will not compile with any as interfaces cannot have non abstract methods.
E. It will compile without filling the blank.

Answer: C
Explanation:
From Java SE 8, we can use static and/or default methods in interfaces, but they should be non abstract methods.
SO in this case using default in blank is completely legal.
Hence option C is correct.
Option A is incorrect as given method is not abstract, so can't use abstract there.
Options B and E are incorrect as we can't have non abstract method interface if they are not default or static.
https;//docs.oraclexom/javase/tutorial/java/Iandl/defaultmethods.html

QUESTION 134
Consider following method

```
default void print(){

        }
```

Which statement is true?

A. This method is invalid.
B. This method can be used only in an interface.
C. This method can return anything.
D. This method can be used only in an interface or an abstract class.
E. None of above.

Answer: B

Explanation:

Given method is declared as default method so we can use it only inside an interface.

Hence option B is correct and option D is incorrect.

Option A is incorrect as it is valid method.

Option C is incorrect as return type is void, which means we can't return anything.

QUESTION 135

Given:

```
public class MyFor3 {
    public static void main(String[] args) {
        int[] xx = null;
        for (int ii : xx) {
            System.out.println(ii);
        }
    }
}
```

What is the result?

A. Null
B. Compilation fails
C. An exception is thrown at runtime
D. 0

Answer: C

QUESTION 136

Given:

```
1.  public class TestLoop {
2.      public static void main(String[] args) {
3.          float myarray[] = {10.20f, 20.30f, 30.40f, 50.60f);
4.          int index = 0;
5.          boolean isFound = false;
6.          float key = 30.40f;
7.          // insert code here
8.          System.out.println(isFound);
9.      }
10. }
```

Which code fragment, when inserted at line 7, enables the code print true?

```
A) while (key == myarray[index++]) {
       isFound = true;
   }

B) while (index <= 4) {
       if (key == myarray[index]) {
       index++;
       isFound = true;
       break;
       }
   }

C) while (index++ < 5) {
       if (key == myarray[index]) {
       isFound = true;
       }
   }

D) while (index < 5) {
       if (key == myarray[index]) {
       isFound = true;
       break;
       }
       index++;
   }
```

A. Option A
B. Option B
C. Option C
D. Option D

Answer: D

QUESTION 137
Given:

```
class Base {
public static void main(String[] args) {
System.out.println("Base " + args[2]);
}
}
public class Sub extends Base{
public static void main(String[] args) {
System.out.println("Overriden " + args[1]);
}
}
And the commands:
javac Sub.java
java Sub 10 20 30
```

What is the result?

A. Base 30
B. Overridden 20
C. Overridden 20
   Base 30
D. Base 30
   Overridden 20

Answer: B

QUESTION 138
Given:

```
class SpecialException extends Exception {
    public SpecialException(String message) {
        super(message);
        System.out.println(message);
    }
}

public class ExceptionTest {
    public static void main(String[] args) {
        try {
            doSomething();
        }
        catch (SpecialException e) {
            System.out.println(e);
        }
    }
    static void doSomething() throws SpecialException {
        int[] ages = new int[4];
        ages[4] = 17;
        doSomethingElse();
    }
    static void doSomethingElse() throws SpecialException {
        throw new SpecialException("Thrown at end of doSomething() method");
    }
}
```

What will be the output?

```
A) SpecialException: Thrown at end of doSomething() method
B) Error in thread "main" java.lang.ArrayIndexOutOfBoundsError
C) Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: 4
    at ExceptionTest.doSomething(ExceptionTest.java:13)
    at ExceptionTest.main(ExceptionTest.java:4)
D) SpecialException: Thrown at end of doSomething() method
    at ExceptionTest.doSomethingElse(ExceptionTest.java:16)
    at ExceptionTest.doSomething(ExceptionTest.java:13)
    at ExceptionTest.main(ExceptionTest.java:4)
```

A. Option A
B. Option B
C. Option C
D. Option D

Answer: C

QUESTION 139
Given the code fragments:

```
interface Contract{ }
class Super implements Contract{ }
class Sub extends Super {}

public class Ref {
    public static void main(String[] args) {
        List objs = new ArrayList();

        Contract c1 = new Super();
        Contract c2 = new Sub();                        // line n1
        Super s1 = new Sub();

        objs.add(c1);
        objs.add(c2);                                   // line n2
        objs.add(s1);

        for(Object itm: objs) {
            System.out.println(itm.getClass().getName());
        }
    }
}
```

What is the result?

A. Super
   Sub
   Sub
B. Contract
   Contract
   Super
C. Compilation fails at line n1
D. Compilation fails at line n2

Answer: A

QUESTION 140
Given:

```
public class Test {
    public static void main(String[] args) {
        Test ts = new Test();
        System.out.print(isAvailable + " ");
        isAvailable= ts.doStuff();
        System.out.println(isAvailable);
    }
    public static boolean doStuff() {
        return !isAvailable;
    }
    static boolean isAvailable = false;
}
```

What is the result?

A. true true
B. true false
C. false true
D. false false
E. Compilation fails

Answer: C