➢ Vendor: Oracle

➢ Exam Code: 1Z0-808

➢ Exam Name: Java SE 8 Programmer I

➢ Question 1 -- Question 20

Visit PassLeader and Download Full Version 1Z0-808 Exam Dumps

QUESTION 1
Given:

```
interface Readable {
    public void readBook();
    public void setBookMark();
}

abstract class Book implements Readable {    // line n1
    public void readBook() { }
    // line n2
}

class EBook extends Book {                   // line n3
    public void readBook() { }
    // line n4
}
```

Which option enables the code to compile?

A) Replace the code fragment at `line n1` with:
```
class Book implements Readable {
```

B) At `line n2` insert:
```
public abstract void setBookMark();
```

C) Replace the code fragment at `line n3` with:
```
abstract class EBook extends Book {
```

D) At `line n4` insert:
```
public void setBookMark() { }
```

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

Answer: D

QUESTION 2
Given the code fragment:
```
public static void main(String[] args) {
    List<String> names = new ArrayList<>();
    names.add("Robb");
    names.add("Bran");
    names.add("Rick");
    names.add("Bran");

    if (names.remove("Bran")) {
        names.remove("Jon");
    }
    System.out.println(names);
}
```
What is the result?

A. [Robb, Rick, Bran]
B. [Robb, Rick]
C. [Robb, Bran, Rick, Bran]
D. An exception is thrown at runtime.

Answer: A
Explanation:
After adding elements to names we have a list with four elements and element “Bran” repeated.
After removing element “Bran” we have a list with three elements [Robb, Rick, Bran].
remove method removes the first occurrence of the specified element from this list, if it is present.
If the list does not contain the element, it is unchanged.
https://docs.oracle.com/javase/8/docs/api/java/util/ArrayList.html#remove-java.lang.Object-

QUESTION 3
Given:

```
class A {
    public A(){
        System.out.print("A ");
    }
}

class B extends A{
    public B(){                          //line n1
        System.out.print("B ");
    }
}

class C extends B{

    public C(){                          //line n2
        System.out.print("C ");
    }
    public static void main(String[] args) {
        C c = new C();
    }
}
```

What is the result?

A. C B A
B. C
C. A B C
D. Compilation fails at line n1 and line n2

Answer: C

QUESTION 4
Given:

```
class X {
    static int i;
    int j;
    public static void main(String[] args) {
        X x1 = new X();
        X x2 = new X();
        x1.i = 3;
        x1.j = 4;
        x2.i = 5;
        x2.j = 6;
        System.out.println(
            x1.i + " " +
            x1.j + " " +
            x2.i + " " +
            x2.j);
    }
}
```

What is the result?

A. 3 4 5 6
B. 3 4 3 6
C. 5 4 5 6
D. 3 6 4 6

Answer: C
Explanation:
Since variable i is static, it is shared by all instances of X. When code executes x2.i = 5, x1.i = 5 too.
Since variable j isn't static, each instance of X has its own copy of j.

QUESTION 5
Given the code fragment:

```
1. public class Test {
2.     public static void main(String[] args) {
3.         /* insert code here */
4.         array[0]=10;
5.         array[1]=20;
6.         System.out.print(array[0]+":"+array[1]);
7.     }
8. }
```

Which code fragment, when inserted at line 3, enables the code to print 10:20?

A. int[] array = new int[2];
B. int[] array;
   array = int[2];
C. int array = new int[2];
D. int array [2] ;

Answer: B

QUESTION 6
Given the code fragment:

```
public static void main(String[] args) {
    String[] arr = {"A", "B", "C", "D"};
    for (int i = 0; i < arr.length; i++) {
        System.out.print(arr[i] + " ");
        if (arr[i].equals("C")) {
            continue;
        }
        System.out.println("Work done");
        break;
    }
}
```

What is the result?

A. A B C Work done
B. A B C D Work done
C. A Work done
D. Compilation fails

Answer: C

QUESTION 7
Which three are advantages of the Java exception mechanism?

A. Improves the program structure because the error handling code is separated from the normal program function
B. Provides a set of standard exceptions that covers all the possible errors
C. Improves the program structure because the programmer can choose where to handle exceptions
D. Improves the program structure because exceptions must be handled in the method in which they occurred
E. Allows the creation of new exceptions that are tailored to the particular program being created

Answer: ACE
Explanation:
B is false. Standard exceptions not cover all possible errors.
D is false. Exceptions don't have to be handled in the method in which they occurred.

QUESTION 8
Given the code from the Greeting.Java file:

```
public class Greeting {
    public static void main(String[] args) {
        System.out.println("Hello " + args[0]);
    }
}
```

Which set of commands prints Hello Duke in the console?

A)
```
javac Greeting
java Greeting Duke
```

B)
```
javac Greeting.java Duke
java Greeting
```

C)
```
javac Greeting.java
java Greeting Duke
```

D)
```
javac Greeting.java
java Greeting.class Duke
```

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

Answer: C
Explanation:
Source code file names must have .java suffixes to compile with javac
We interpret or run the program with “java <class name without suffix> arguments”
http://docs.oracle.com/javase/8/docs/technotes/tools/windows/javac.html
http://docs.oracle.com/javase/8/docs/technotes/tools/windows/java.html

QUESTION 9
Given:

```
class Alpha {
    int ns;
    static int s;
    Alpha(int ns) {
        if (s < ns) {
            s = ns;
            this.ns = ns;
        }
    }
    void doPrint() {
        System.out.println("ns = " + ns + " s = " + s);
    }
}
```

And,

```
public class TestA {
    public static void main(String[] args) {
        Alpha ref1 = new Alpha(50);
        Alpha ref2 = new Alpha(125);
        Alpha ref3 = new Alpha(100);
        ref1.doPrint();
        ref2.doPrint();
        ref3.doPrint();
    }
}
```

What is the result?

A)
```
ns = 50 s = 125
ns = 125 s = 125
ns = 100 s = 125
```

B)
```
ns = 50 s = 125
ns = 125 s = 125
ns = 0 s = 125
```

C)
```
ns = 50 s = 50
ns = 125 s = 125
ns = 100 s = 100
```

D)
```
ns = 50 s =  50
ns = 125 s = 125
ns = 0 s = 125
```

A. Option A
B. Option B

C. Option C
D. Option D

Answer: B
Explanation:
After ref1 is instantiated, ref1.ns = 50 and s = 50
After ref2 is instantiated, ref2.ns = 125 and s = 125
After ref3 is instantiated, ref3.ns = 0 and s = 125

QUESTION 10
Given the code fragment:

```
public static void main(String[] args) {
    int ii = 0;
    int jj = 7;
    for (ii = 0; ii < jj - 1; ii = ii + 2) {
        System.out.print(ii + " ");
    }
}
```

What is the result?

A. 2 4
B. 0 2 4 6
C. 0 2 4
D. Compilation fails

Answer: C

QUESTION 11
Given the code fragment:

```
LocalDate date1 = LocalDate.now();
LocalDate date2 = LocalDate.of(2014, 6, 20);
LocalDate date3 = LocalDate.parse("2014-06-20", DateTimeFormatter.ISO_DATE);
System.out.println("date1 = " + date1);
System.out.println("date2 = " + date2);
System.out.println("date3 = " + date3);
```

Assume that the system date is June 20, 2014. What is the result?

A) date1 = 2014-06-20
date2 = 2014-06-20
date3 = 2014-06-20

B) date1 = 06/20/2014
date2 = 2014-06-20
date3 = Jun 20, 2014

C) Compilation fails.

D) A DateParseExcpetion is thrown at runtime.

A. Option A

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

Answer: A
Explanation:
I've run the following code without any problem
import java.time.LocalDate;
import java.time.format.DateTimeFormatter;
public class Main {
public static void main(String[] args) {
LocalDate date1 = LocalDate.now();
LocalDate date2 = LocalDate.of(2014, 6, 20);
LocalDate date3 = LocalDate.parse("2014-06-20", DateTimeFormatter.ISO_DATE);
System.out.println("date 1 = " + date1);
System.out.println("date 2 = " + date2);
System.out.println("date 3 = " + date3);
}
}
The output is
date 1 = 2015-09-05 (because run today, but problem statement says we must assume that the system data is June 20, 2014)
date 2 = 2014-06-20
date 3 = 2014-06-20

QUESTION 12
Given the code fragment:

```
7.   StringBuilder sb1 = new StringBuilder("Duke");
8.   String str1 = sb1.toString();
9.   // insert code here
10.    System.out.print(str1 == str2);
```

Which code fragment, when inserted at line 9, enables the code to print true?

A. String str2 = str1;
B. String str2 = new String (str1);
C. String str2 = sb1. toString ();
D. String str2 = "Duke";

Answer: A
Explanation:
Operator == checks if two things are EXACTLY the same thing, not if they have the same content

QUESTION 13
Given the code fragment:

```
public class Test {

    static int count = 0;
    int i = 0;

    public void changeCount() {
        while (i < 5) {
            i++;
            count++;
        }
    }

    public static void main(String[] args) {
        Test check1 = new Test();
        Test check2 = new Test();
        check1.changeCount();
        check2.changeCount();
        System.out.print(check1.count + " : " + check2.count);
    }
}
```

What is the result?

A. 10 : 10
B. 5 : 5
C. 5 : 10
D. Compilation fails

Answer: A
Explanation:
The variable i is local to all instances of class Test so each time we create an instance, i=0 and the loop add 5 to count.
The variable count (static) is global to all instances of class Test and all instances share the same variable. It's been initialized only once to zero an retains its value between the calls to changeCount
Since we call two times the method changeCount, the final result is 10 : 10

QUESTION 14
Given the code fragment:

```
public static void main(String[] args) {
    double discount = 0;
    int qty = Integer.parseInt(args[0]);
    //line n1;
}
```

And given the requirements:
- If the value of the qty variable is greater than or equal to 90, discount = 0.5
- If the value of the qty variable is between 80 and 90, discount = 0.2

Which two code fragments can be independently placed at line n1 to meet the requirements?

☐ A)
```
if (qty >= 90) { discount = 0.5; }
if (qty > 80 && qty < 90) { discount = 0.2; }
```

☐ B)
```
discount = (qty >= 90) ? 0.5 : 0;
discount = (qty > 80) ? 0.2 : 0;
```

☐ C)
```
discount = (qty >= 90) ? 0.5 : (qty > 80)? 0.2 : 0;
```

☐ D)
```
if (qty > 80 && qty < 90) {
    discount = 0.2;
} else {
    discount = 0;
}
if (qty >= 90) {
    discount = 0.5;
} else {
    discount = 0;
}
```

☐ E)
```
discount = (qty > 80) ? 0.2 : (qty >= 90) ? 0.5 : 0;
```

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

Answer: AC

QUESTION 15
Given:

```
public class Test {

    public static void main(String[] args) {
        if (args[0].equals("Hello") ? false : true) {
            System.out.println("Success");
        } else {
            System.out.println("Failure");
        }
    }
}
```

And given the commands:

```
javac Test.Java
Java Test Hello
```

What is the result?

A. Success

B. Failure
C. Compilation fails.
D. An exception is thrown at runtime

Answer: B

QUESTION 16
Which three statements describe the object-oriented features of the Java language?

A. Objects cannot be reused.
B. A subclass can inherit from a superclass.
C. Objects can share behaviors with other objects.
D. A package must contain more than one class.
E. Object is the root class of all other objects.
F. A main method must be declared in every class.

Answer: BCE
Explanation:
https://docs.oracle.com/javase/tutorial/java/IandI/subclasses.html
http://www.artima.com/objectsandjava/webuscript/PolymorphismInterfaces1.html

QUESTION 17
Given the following code:

```
public static void main(String[] args){
    String[] planets = {"Mercury", "Venus", "Earth", "Mars"};

    System.out.println(planets.length);
    System.out.println(planets[1].length());
}
```

What is the output?

A. 4
   4
B. 3
   5
C. 4
   7
D. 5
   4
E. 4
   5
F. 4
   21

Answer: E

QUESTION 18
You are developing a banking module.
You have developed a class named ccMask that has a maskcc method.
Given the code fragment:

```
class CCMask {
    public static String maskCC(String creditCard) {
        String x = "XXXX-XXXX-XXXX-";
        //line n1
    }

    public static void main(String[] args) {
        System.out.println(maskCC("1234-5678-9101-1121"));
    }
}
```

You must ensure that the maskcc method returns a string that hides all digits of the credit card number except the four last digits (and the hyphens that separate each group of four digits).
Which two code fragments should you use at line n1, independently, to achieve this requirement?

☐ A)
```
StringBuilder sb = new StringBuilder(creditCard);
sb.substring(15, 19);
return x + sb;
```

☐ B)
```
return x + creditCard.substring(15, 19);
```

☐ C)
```
StringBuilder sb = new StringBuilder(x);
sb.append(creditCard, 15, 19);
return sb.toString();
```

☐ D)
```
StringBuilder sb = new StringBuilder(creditCard);
StringBuilder s = sb.insert(0, x);
return s.toString();
```

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

Answer: BC

QUESTION 19
Given the code fragment:

```
public class App {
    public static void main(String[] args) {
        String str1 = "Java";
        String str2 = new String("java");
        //line n1
        {
            System.out.println("Equal");
        } else {
            System.out.println("Not Equal");
        }
    }
}
```

Which code fragment, when inserted at line n1, enables the App class to print Equal?

A)
```
String str3 = str2;
if (str1 == str3)
```

B)
```
if (str1.equalsIgnoreCase(str2))
```

C)
```
String str3 = str2;
if (str1.equals(str3))
```

D)
```
if (str1.toLowerCase() == str2.toLowerCase())
```

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

Answer: A

QUESTION 20
Given:

```
public class SumTest {

    public static void doSum(Integer x, Integer y) {
        System.out.println("Integer sum is " + (x + y));
    }

    public static void doSum(double x, double y) {
        System.out.println("double sum is " + (x + y));
    }

    public static void doSum(float x, float y) {
        System.out.println("float sum is " + (x + y));
    }

    public static void doSum(int x, int y) {
        System.out.println("int sum is " + (x + y));
    }

    public static void main(String[] args) {
        doSum(10, 20);
        doSum(10.0, 20.0);
    }
}
```

What is the result?

A)
```
int sum is 30
float sum is 30.0
```

B)
```
int sum is 30
double sum is 30
```

C)
```
Integer sum is 30
double sum is 30.0
```

D)
```
Integer sum is 30
float sum is 30.0
```

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

Answer: B
Explanation:
int is a primitive type and Integer is an object with an int. When we call doSum(10, 20), we are calling doSum(int, int).

By default, Java use double to represent its floating point literals. When we call doSum(10.0, 20.0), we are calling doSum(double, double).

Visit PassLeader and Download Full Version 1Z0-808 Exam Dumps