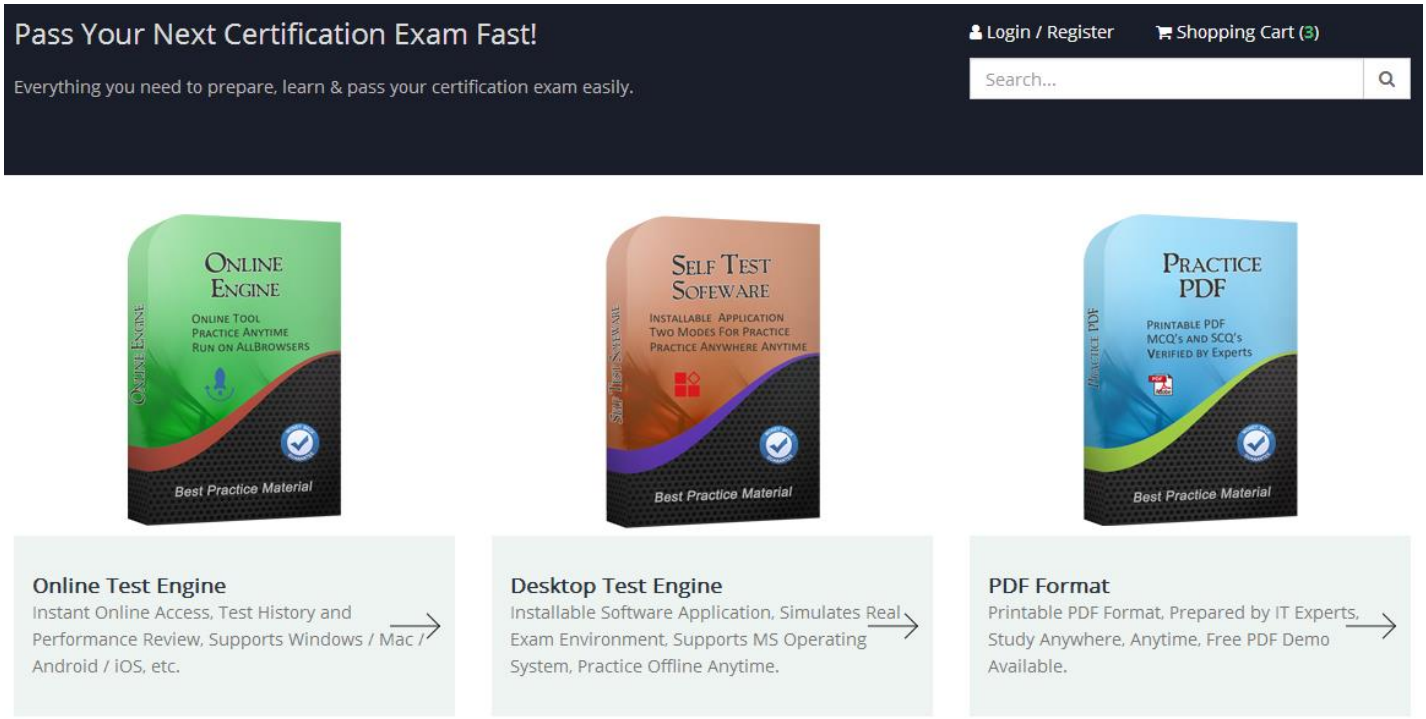
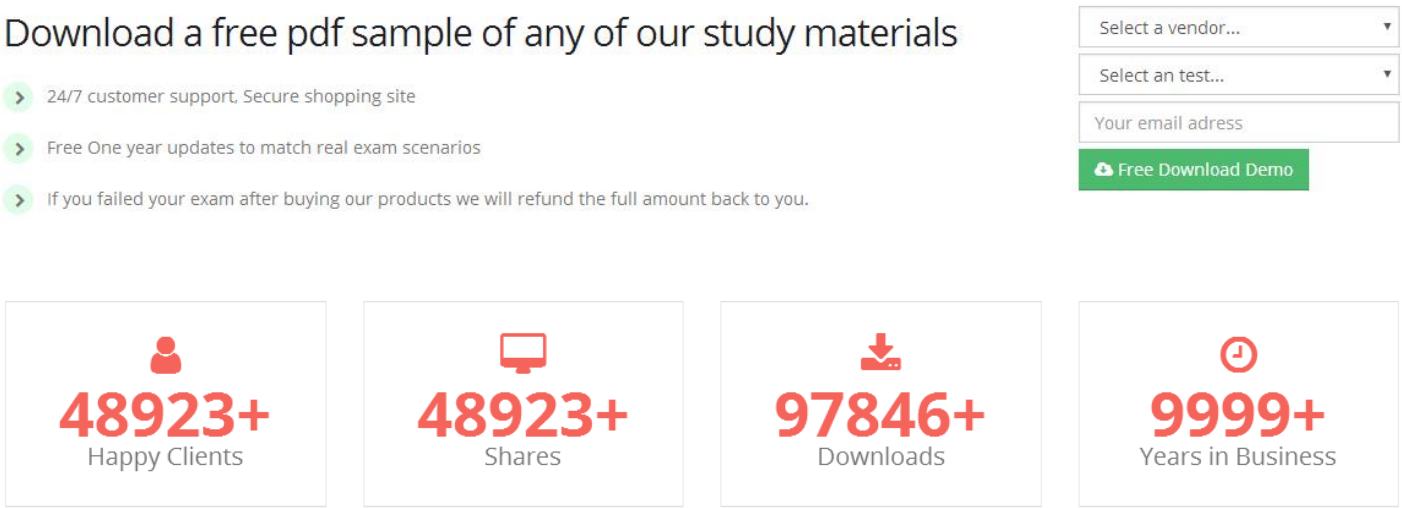
iPassLeader





http://www.ipassleader.com/

Everything you need to prepare, learn & pass your certification exam easily.

Exam : **1z0-816**

Title : Java SE 11 Programmer II

Vendor : Oracle

Version : DEMO

NO.1 Which three annotation uses are valid? (Choose three.)
A. Function<String, String> func = (@NonNull x) -> x.toUpperCase();
B. var v = "Hello" + (@Interned) "World"
C. var myString = (@NonNull String) str;
D. var obj = new @Interned MyObject();
E. Function<String, String> func = (@NonNull var x) -> x.toUpperCase();
F. Function<String, String> func = (var @NonNull x) -> x.toUpperCase();
Answer: A,D,F

NO.2 Which code fragment prints 100 random numbers?

A.
```
var r=  new Random();
new DoubleStream(r::nextDouble).limit(100).forEach(System.out::print);
```
B.
```
DoubleStream.generate(Random::nextDouble)
            .limit (100).forFach(System.out::print);
```
C.
```
Doublestream.generate(Random.nextDouble).limit(100).forEach(System.out.print);
```
D.
```
var r = new Random(); DoubleStream.generate(r::nextDouble).limit(100).forEach(System.out::print);
```

A. Option A
B. Option D
C. Option C
D. Option B
Answer: B
Reference:
https://www.javacodegeeks.com/2014/01/java-util-random-in-java-8.html

NO.3 Which is a proper JDBC URL?
A. http://localhost.mysql.com:3306/database
B. jdbc:mysql://localhost:3306/database
C. http://localhost mysql.jdbc:3306/database
D. jdbe.mysql.com://localhost:3306/database
Answer: B
Reference:
https://vladmihalcea.com/jdbc-driver-connection-url-strings/

NO.4 Given:
List<String> longlist = List.of("Hello","World","Beat");
List<String> shortlist = new ArrayList<>();
Which code fragment correctly forms a short list of words containing the letter "e"?

A.
```
longList.stream()
        .filter(w -> w.indexOf('e') != -1)
        .parallel()
        .forEach(w -> shortList.add(w));
```

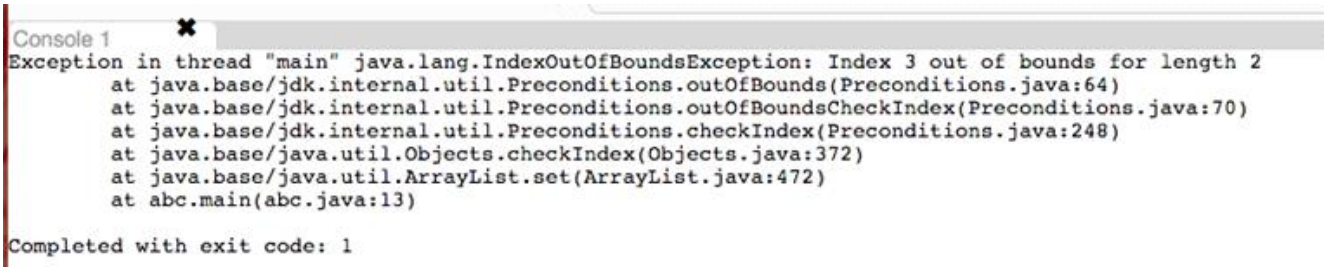
B.
```
longList.parallelStream()
        .filter(w -> w.indexOf('e') != -1)
        .forEach(w -> shortList.add(w));
```

C.
```
shortList = longList.stream()
        .filter(w -> w.indexOf('e') != -1)
        .parallel()
        .collect(Collectors.toList());
```

D.
```
longList.stream()
        .filter(w -> w.indexOf('e') != -1)
        .parallel()
        .collect(shortlist);
```

A. Option B
B. Option A
C. Option D
D. Option C
Answer: D

NO.5 Given:
var data = new ArrayList<>();
data.add("Peter");
data.add(30);
data.add("Market Road");
data.set(1, 25);
data.remove(2);
data.set(3, 1000L);
System.out.print(data);
What is the output?
A. [Peter, 30, Market Road]
B. An exception is thrown at run time.
C. [Peter, 25, null, 1000]
D. [Market Road, 1000]
Answer: B
Explanation:



NO.6 Given:

```
public class Confidential implements Serializable{
   private String data;

   public Confidential(String data) {
      this.data = data;
   }
}
```

Which two are secure serialization of these objects? (Choose two.)

A. Implement only writeReplace to replace the instance with a serial proxy and not readResolve.

B. Define the serialPersistentFields array field.

C. Implement only readResolve to replace the instance with a serial proxy and not writeReplace.

D. Declare fields transient.

E. Make the class abstract.

Answer: B,C

NO.7 Given the code fragment:

var pool = Executors.newFixedThreadPool(5);

Future outcome = pool.submit(() -> 1);

Which type of lambda expression is passed into submit()?

A. java.util.function.Predicate

B. java.util.function.Function

C. java.lang.Runnable

D. java.util.concurrent.Callable

Answer: D

Reference:

https://www.codota.com/code/java/methods/java.util.concurrent.Executors/newFixedThreadPool

NO.8 Given:

```
public class Employee {
   private String name;
   private String locality;
   /* the constructor, getter and setter methods code goes here */
}
```

and:

```
 8. List<Employee> roster = new ArrayList<>();
 9. long empCount = roster.stream()
10. /* insert code here */
11. System.out.print(empCount);
```

Which code, when inserted on line 10, prints the number of unique localities from the roster list?

A. map(e -> e.getLocality())
.count();

B. .filter(Employee::getLocality)
.distinct()
.count();

C. .map(e -> e.getLocality())
.collect(Collectors.toSet())
.count();

D. .map(Employee::getLocality)
.distinct()
.count();

Answer: B

Reference:
https://developer.android.com/reference/android/location/Address