

Exam 813 – Sample questions

1. Which code fragment correctly assign a numeric literal?

- A) `byte b1 = b1011;`
- B) `byte b2 = 1011b;`
- C) `byte b3 = 0b1001;`
- D) `byte b4 = 0xb001;`

2. Given the fragment:

```
public class MathFun {  
    public static void main(String[] args) {  
        int number1 = 0b0111;  
        int number2 = 0111_000;  
        System.out.println("Number1: " + number1);  
        System.out.println("Number2: " + number1);  
    }  
}
```

What is the result?

- A) Number1: 7
Number2: 7
- B) Number1: 7
Number2: 111_000
- C) Number1: 0b0111
Number2: 0111000
- D) Compilation fails.

3. Given:

The `test1.txt` file is available and the `test2.txt` file is not available.

And, given the code fragment:

```
Path sPath = Paths.get("test1.txt");
Path dPath = Paths.get("test2.txt");
try {
    Files.move(sPath, dPath, StandardCopyOption.ATOMIC_MOVE);
} catch (IOException ex) {
    System.err.println("Exception!");
}
```

Which statement is true?

- A) The `test1.txt` file is renamed to the `test2.txt` file and the `test1.txt` file is removed in a single operation.
- B) The `test1.txt` file is renamed to the `test2.txt` file and the `test1.txt` file is removed in two distinct operation.
- C) The `test1.txt` file is copied and renamed to the `test2.txt` file in a single operation.
- D) The program prints: `Exception!`

4. Which is a valid functional interface?

- A)

```
public interface Useful<E> {
    E getStuff();
    void putStuff(E e);
}
```
- B)

```
public interface Useful {
    void doStuff();
    default void doOtherStuff() {}
}
```
- C)

```
@FunctionalInterface
public interface Useful{ default void doStuff(){} }
```
- D)

```
public interface Useful {
    abstract void doStuff();
    abstract void doCalc();
}
```

5. Given the code fragment:

```
public class App {  
    public static void main(String[] args) {  
        String s = "Java";  
        String n = "SE";  
        // Line n1  
        System.out.println(sf.apply(s, n));  
    }  
}
```

Which code fragment, when inserted at Line n1, prints JavaSE?

- A) BiFunction<String, String, String> sf = (s1, n1) -> s1.concat(n1);
- B) BiFunction<String, String> sf = (s1, n1) -> s1.concat(n1);
- C) Function<String, String> sf = (s1, n1) -> s1.concat(n1);
- D) Function<String, String, String> sf = (s1, n1) -> s1.concat(n1);

6. Given the code fragment:

```
Arrays.asList("Fred", "Jim", "Sheila")  
    .stream()  
    .peek(System.out::println)  
    .allMatch(s -> s.startsWith("F"));
```

What is the result?

- A) Fred
Jim
Sheila
- B) Fred
Jim
- C) Fred
- D) Compilation fails.

7. Given the code fragment:

```
LocalDate date1 = LocalDate.of(2016, Month.JANUARY, 1);  
LocalDateTime date2 = LocalDateTime.of(2017, Month.JUNE, 1, 1, 1);  
Period p = Period.between(date1, date2);  
System.out.print(p.getYears() + ":" + p.getMonths() + ":" + p.getDays());
```

What is the result?

- A) 1:5:0
- B) 1:6:0
- C) 0:0:0
- D) Compilation fails.

8. Given the code fragment:

```
class MyResource1 implements Closeable {  
    public void close() {  
        System.out.print("r1 ");  
    }  
}  
  
class MyResource2 implements AutoCloseable {  
    public void close() throws IOException {  
        System.out.print("r2 ");  
        throw new IOException();  
    }  
}  
  
public class App2 {  
    public static void main(String[] args) {  
        try (MyResource1 r1 = new MyResource1();  
            MyResource2 r2 = new MyResource2();) {  
            System.out.print("try ");  
        } catch (Exception e) {  
            System.out.print("catch ");  
            for (Throwable t : e.getSuppressed()) {  
                System.out.println(t.getClass().getName());  
            }  
        }  
    }  
}
```

What is the result?

- A) try r2 r1 catch java.io.IOException
- B) try r2 r1 catch
- C) try r1 r2 catch
- D) Compilation fails.

Answers:

- 1. C
- 2. A
- 3. A
- 4. B
- 5. A
- 6. B
- 7. D
- 8. B