Simulation Exam Java Basic 2022-2023

Professeur : M. BOULCHAHOUB HASSAN Durée : 1 h 00 min

*Required

1. Nom & Prénom :

2. Donner l'affichage qui résulte de l'exécution de la classe Test Suivante *

```
package ma.cigma.exam.basic.q6.blocinit;
public class Test {
    public static void main(String[] args) {
        Processing p1=new Processing( time: 10);
        Processing p2=new Processing( time: 20);
    }
}
```

```
public class Processing {
    private long time;
    private static String job="Job";
    {
        time += 10;
        System.out.println("A="+time);
    }
    static {
        job +="1";
        System.out.println("B= "+job);
    }
    {
        time += 30;
        System.out.println("C "+time);
    }
    public Processing(long time) {
        this.time += time;
        System.out.println("D "+this.time);
    }
}
```

---

3. Donner le résultat des l'expression Lambda de la classe Test suivante: *

```
package ma.cigma.exam.basic.q8.lambda;

import java.util.Arrays;
import java.util.List;

public class Test {
    public static void main(String[] args) {
        List<Integer> numbers= Arrays.asList(2,3,4);
        numbers.stream().map(i→3*i).map(i→i*i).filter(i→i%2==0)
                .forEach(i→ System.out.println(i));
    }
}
```

---

4. Donner l'affichage qui va résulter de l'exécution de la classe Test suivante: *

```java
package ma.cigma.exam.basic.q7.varargs;

public class Test {
    public static void main(String[] args) {
        Compute.process( a: 10, b: 20);
    }
}
```

```java
public class Compute {
    static void process(Integer a,Integer b){
        System.out.println("process Integer");
    }
    static void process(long a,long b){
        System.out.println("process long");
    }
    static void process(int... a){
        System.out.println("process int...");
    }
}
```

5. Quels sont les objets row qui seront stockés dans la collection rows *

```
import java.util.Set;

public class Test {
    public static void main(String[] args) {
        Set<Row> rows = new HashSet<>();
        rows.add(new Row( number: 3, content: "FIX"));
        rows.add(new Row( number: 2, content: "ISSUE"));
        rows.add(new Row( number: 2, content: "MERGE"));
        rows.add(new Row( number: 3, content: "PULL"));
        rows.add(new Row( number: 3, content: "IOC"));
        System.out.println(rows);
    }
}
```

```
public class Row {
    private int number;
    private String content;
    @Override
    public boolean equals(Object o) {
        Row row = (Row) o;
        return content.length()==row.content.length();
    }
    @Override
    public int hashCode() {
        return number+content.length();
    }
    public Row(int number, String content) {
        this.number = number;
        this.content = content;
    }
    @Override
    public String toString() {
        return "Row{number="+ number +": content=" +content+ "}";
    }
}
```

Tick all that apply.

☐ [Row{number=3: content=IOC}, Row{number=2: content=ISSUE}, Row{number=3: content=PULL}]

- [ ] [Row{number=3: content=FIX}, Row{number=2: content=MERGE}, Row{number=3: content=PULL}]
- [ ] [Row{number=3: content=FIX}, Row{number=2: content=ISSUE}, Row{number=3: content=PULL}]
- [ ] [Row{number=3: content=FIX}, Row{number=2: content=ISSUE}]

6. Donner l'affichage de la classe Test suivante *

```java
package ma.cigma.exam.basic.q9.comparator;

import java.util.Set;
import java.util.TreeSet;

public class Test {
    public static void main(String[] args) {
        Set<String> identifiers= new TreeSet<>((a,b)->a.length()-b.length());
        identifiers.add("A200");
        identifiers.add("A400");
        identifiers.add("A300");
        identifiers.forEach(a->System.out.println(a));
    }
}
```
