CSA0961 – JAVA

TEST – 6 (SET – B)

1.public class Counter {

private int count = 0;

public void increment() {

count++;

}

public int getCount() {

return count;

}

}

public class Test {

public static void main(String[] args) {

Counter counter = new Counter();

while (counter.getCount() < 10) {

counter.increment();

}

System.out.println("Counter reached: " + counter.getCount());

}

}

 Issue: Static field not retaining value across instances.

 Solution: Check singleton implementation for proper instance handling.

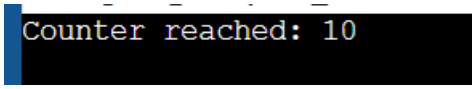
DEBUGGING :

public class Main {

```
    public static class Counter {
        private int count = 0;

        public void increment() {
            count++;
```

```
    }

    public int getCount() {
        return count;
    }
  }

  public static void main(String[] args) {
    Counter counter = new Counter();

    while (counter.getCount() < 10) {
        counter.increment();
    }

    System.out.println("Counter reached: " + counter.getCount());
  }
}
```

OUTPUT :



2. public class Employee {

private String name;

public Employee(String name) {

this.name = name;

}

public String getName() {

return name;

}

}

public class Test {

public static void main(String[] args) {

Employee e = new Employee("John");

System.out.println(e.name); // Compilation error

Issue: Direct access to private field name.

 Solution: Use getter method getName() to access private fields.

DEBBUGING :

```
class Employee
{
 String name;
public Employee(String name)
{
this.name = name;
}
public String getName() {
return name;
 }
}
public class Main
{
public static void main(String[] args)
{
Employee e = new Employee("John");
System.out.println(e.name);
}
}
```

OUTPUT :

```
John
```

4. Question: Why is my array not printing the correct values?

 Potential Issue: Ensure the array values are set correctly before printing.

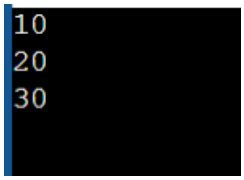
```
public class PrintArray {

public static void main(String[] args) {
int[] numbers = new int[3];
```

```
numbers[0] = 10;
numbers[1] = 20;
numbers[2] = 30;
for (int num : numbers) {
System.out.println(num);
```

DEBUGGING :

```
public class Main
{
public static void main(String[] args)
{
int[] numbers = new int[3];
numbers[0] = 10;
numbers[1] = 20;
numbers[2] = 30;
for (int num : numbers)
{
System.out.println(num);
}
}
}
```

OUTPUT :



3. Question: Why is the FileNotFoundException not being caught when trying to open a file?

 Potential Issue: Make sure the FileInputStream or FileReader is enclosed in a try-catch block.

```
public class FileOpener {
public void openFile(String filePath) {
try {
```

```
FileReader fileReader = new FileReader(filePath);
BufferedReader br = new BufferedReader(fileReader);
String line;
while ((line = br.readLine()) != null) {
System.out.println(line);
}
br.close();
} catch (FileNotFoundException e) {
System.out.println(&quot;File not found: &quot; + filePath);
} catch (IOException e) {
e.printStackTrace();
}
}
}

public class TestFileOpener {
public static void main(String[] args) {
FileOpener opener = new FileOpener();
opener.openFile(&quot;missingfile.txt&quot;);
```

DEBBUGGING :

```
import java.io.BufferedReader;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.IOException;

class FileOpener {
    public void openFile(String filePath) {
        try {
            FileReader fileReader = new FileReader(filePath);
            BufferedReader br = new BufferedReader(fileReader);
            String line;
            while ((line = br.readLine()) != null) {
```

```
            System.out.println(line);
        }
        br.close();
    } catch (FileNotFoundException e) {
        System.out.println("File not found: " + filePath);
    } catch (IOException e) {
        e.printStackTrace();
    }
  }
}

public class Main {
  public static void main(String[] args) {
    FileOpener opener = new FileOpener();
    opener.openFile("missingfile.txt");
  }
}
```

OUTPUT :

```
File not found: missingfile.txt
```