**AIM: Apply Singleton Design Pattern on the Domain Chosen**

**Domain**: **Cybersecurity**, Usage: Setting Up a Company’s BaseDetector to avoid Email Spamming.

|  |
| --- |
| //creating a class that will be used to create a singleton object and then any other object that is created will be a clone of the singleton object  // singleton Design Pattern  public class baseDetector {      private static baseDetector instance = null;      private String name;      private String type;      private baseDetector() {}      public static baseDetector getInstance() {          if (instance == null) {              instance = new baseDetector();          }          return instance;      }      public String getName() {          return name;      }      public void setName(String name) {          this.name = name;      }      public String getType() {          return type;      }      public void setType(String type) {          this.type = type;      }      public static void main(String[] args) {          baseDetector baseDetector1 = baseDetector.getInstance();          baseDetector1.setName("John Doe Company's Base Detector");          baseDetector1.setType("Email");          System.out.println("Name of baseDetector1: " + baseDetector1.getName());          System.out.println("Type of baseDetector1: " + baseDetector1.getType());          baseDetector baseDetector2 = baseDetector.getInstance();          System.out.println("Name of baseDetector2: " + baseDetector2.getName());          System.out.println("Type of baseDetector2: " + baseDetector2.getType());  }  } |

**Output:**

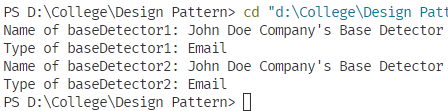
****

Figure 1: Output of the above program