**Exercise 1: Implementing the Singleton Pattern**  
  
Solution Code:

**class** Logger {

**private** **static** Logger instance;

**private** Logger() {

System.out.println("Initialized Logger instance");

}

**public** **static** Logger getInstance() {

if (instance == null) {

instance = new Logger();

}

return instance;

}

**public** **void** log(String message) {

System.out.println("Log Message: " + message);

}

}

**class** Tester {

**public** **static** **void** main(String[] args) {

Logger logger1 = Logger.getInstance();

Logger logger2 = Logger.getInstance();

logger1.log("First message");

logger2.log("Second message");

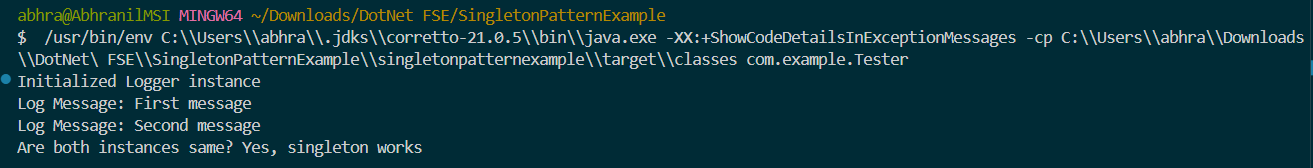
System.out.println(

"Are both instances same? " + ((logger1 == logger2) ? "Yes, singleton works" : "No, singleton failed"));

}

}

Output:



**Exercise 2: Implementing the Factory Method Pattern**

Solution Code:

**interface** Document {

**void** create();

}

**class** WordDocument **implements** Document {

**public** **void** create() {

System.out.println("Creating Word document");

}

}

**class** PdfDocument **implements** Document {

**public** **void** create() {

System.out.println("Creating PDF document");

}

}

**class** ExcelDocument **implements** Document {

**public** **void** create() {

System.out.println("Creating Excel document");

}

}

**public** **class** DocumentFactory {

**public** Document createDocument(String type) {

switch (type.toLowerCase()) {

case "excel":

return new ExcelDocument();

case "word":

return new WordDocument();

case "pdf":

return new PdfDocument();

default:

throw new IllegalArgumentException("Unknown document type: " + type);

}

}

}

**public** **class** Tester {

**public** **static** **void** main(String[] args) {

DocumentFactory factory = new DocumentFactory();

Document word = factory.createDocument("word");

word.create();

Document excel = factory.createDocument("excel");

excel.create();

Document pdf = factory.createDocument("pdf");

pdf.create();

Document unknown = factory.createDocument("unknown"); *// This will throw an exception*

unknown.create();

}

}

Output:  
