Name: Chiranjivi R
Contact: +91 9513030949
Email: chiranjivir23is@rnsit.ac.in

Assignment 1 – Java Code Review and Error Correction

Objective: Identify, correct, and explain errors in Java code snippets to strengthen understanding of Java syntax and structure.

Task1: Review the following codes, find and fix errors also explain the errors

1. **Code Snippet**

   GIVEN CODE:

```
public class Main {
        public static void main(String[] args) {
                System.out.println("Hello, World!")
                                                }
                                }
```

   Error:

   Missing semicolon ; at the end of the println statement.

   Corrected Code:

```
public class Main {
   public static void main(String[] args) {
      System.out.println("Hello, World!");  }}
```

   Explanation:

   Every Java statement must end with a semicolon. Without it, the code will not compile.

2. Code Snippet

GIVEN CODE:

```
public class Main {
        public void greet() {
                System.out.println("Hello");
                }
        greet(); }
```

Error:

greet() is called directly inside the class, not inside a method.

Also, **greet()** is non-static but called from static context.

Corrected Code:

```
public class Main {

   public static void main(String[] args) {
      greet();
   }

   public static void greet() {
      System.out.println("Hello");
   }
}
```

Explanation:

Method calls must be inside another method.

Static methods can be called directly from main, so **greet()** must also be static.

3. Code Snippet

GIVEN CODE:

```
public class Main {
        public static void main(String[] args) {
                int number = "10";
                System.out.println("The number is: " + number);
                                                        }

                }
```

Error:

Assigning a string **("10")** to an integer variable.

Corrected Code:

```
public class Main {
   public static void main(String[] args) {
      int number = 10;
      System.out.println("The number is: " + number);
                                                    }
            }
```

Explanation:

int can only store numeric values, not strings.

4. Code Snippet

GIVEN CODE:

```
public class Main {
        public static void main(String[] args) {
                int[] numbers = {1, 2, 3, 4};
                System.out.println("The fifth element is: " + numbers[4]);
                                }
        }
```

Error:

Trying to access index 4 in an array of size 4.

Array indices start from 0 to 3.

Corrected Code:

```
public class Main {

   public static void main(String[] args) {

      int[] numbers = {1, 2, 3, 4};

      System.out.println("The fourth element is: " + numbers[3]);

   }

}
```

Explanation:

The array has 4 elements, so valid indices are 0 to 3.

Trying to access numbers[4] will cause an **ArrayIndexOutOfBoundsException** because the 5th element does not exist.

The correct way is to use numbers[3] to access the fourth element.

5. Code Snippet

GIVEN CODE:

```
public class Main {
        public static void main(String[] args) {
                int result = addNumbers(5, 10);
                System.out.println("Result: " + result);
        }
public int addNumbers(int a, int b) {
                return a + b;
                }
}
```

Error:

Calling a non-static method **(addNumbers)** from static main.

Corrected Code:

```
public class Main {
   public static void main(String[] args) {
      int result = addNumbers(5, 10);
      System.out.println("Result: " + result);
   }

   public static int addNumbers(int a, int b) {
return a + b;
                                                }

        }
```

Explanation:

Static methods must call only static methods unless you create an object.

6. Code Snippet

GIVEN CODE:

```
public class Main {
        public static void main(String[] args) {
                int age;
                if (age >= 18) {
                        System.out.println("You are eligible to vote.");
                                }
                                                }
        }
```

Error:

Variable age is declared but not initialized.

Corrected Code:

```
public class Main {
   public static void main(String[] args) {
      int age = 18;

      if (age >= 18) {
         System.out.println("You are eligible to vote.");
      }
   }
}
```

Explanation:

Local variables must be assigned a value before use.

7. Code Snippet

Error:

Variable i is not accessible outside the for loop (out of scope).

Corrected Code:

```
public class Main {
   public static void main(String[] args) {
      int i;
      for (i = 0; i < 5; i++) {
         System.out.println("Number: " + i);
      }
      System.out.println("Outside loop: " + i);
   }
}
```

Explanation:

Declaring i before the loop makes the variable accessible outside the loop.

8. Code Snippet

GIVEN CODE:

```
public class Main {
        public static void main(String[] args) {
        while count < 10 {
                        System.out.println("Count: " + count);
            count++;
                                }
                                                }
        }
```

Error:

Missing parentheses in while

count is not declared

Braces cannot be used because the while syntax is invalid

Corrected Code:

```
public class Main {
    public static void main(String[] args) {
        int count = 0;

        while (count < 10) {
            System.out.println("Count: " + count);
            count++;
        }
    }
}
```

Explanation:

The correct syntax is while(condition) { }.

Variables must be declared before use.