Section 1: Error-Driven Learning in Java

Snippet 1:

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

Error :-

---------> The declaration of main method is wrong. 'static' is used to call method without creating instance of class.

Corrected Code Snippet:-

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

Snippet 2:

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

Error:-

---------> The declaration of main method is wrong. 'public' is used to access method by java runtime system.

Corrected Code Snippet:-

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

Snippet 3:

```
public class Main {
    public static int main(String[] args) {
        System.out.println("Hello, World!");
        return 0;
    }
 }
```

Error:-

---------> The declaration of main method is wrong. 'int' should be replaced with 'void' so that the method is non returning type.

Corrected Code Snippet:-

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

Snippet 4:

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

Error:-

---------> The declaration of main method is wrong. 'String[] args' is used for command line arguments.

Corrected Code Snippet:-

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

Snippet 5:

```
public class Main {
    public static void main(String[] args) {
        System.out.println("Main method with String[] args");
    }
    public static void main(int[] args) {
        System.out.println("Overloaded main method with int[] args");
    }
}
```

Error:-

---------> The main method is overloaded. But java runtime machine will not call second main method as only first method will be considered as entry point. Hence in order to have second method to run, call that overloaded method.

Corrected Code Snippet:-

```
public class Main {
    public static void main(String[] args) {
        System.out.println("Main method with String[] args");

        int[] intArray = {1, 2, 3};
        main(intArray);
    }

    public static void main(int[] args) {
        System.out.println("Overloaded main method with int[] args");
    }
}
```

Snippet 6:

```
public class Main {
    public static void main(String[] args) {
        int x = y + 10;
        System.out.println(x);
    }
}
```

Error:-

---------> the variable 'y' is not declared and directly used. To use it we need to declare it along with initial value.

Corrected Code Snippet:-

```
public class Main {
    public static void main(String[] args) {
        int y = 5;
        int x = y + 10;
        System.out.println(x);
    }
}
```

Snippet 7:

```
public class Main {
    public static void main(String[] args) {
        int x = "Hello";
        System.out.println(x);
    }
}
```

Error:-

---------> 'int' cannot store array of characters. Hence use 'String' datatype for array of characters.

Corrected Code Snippet:-

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

Snippet 8:

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

Error:-

---------> The print statement is missing a parenthesis and a semicolon. It is must to follow java syntax.

Corrected Code Snippet:-

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

Snippet 9:

```
public class Main {
    public static void main(String[] args) {
        int class = 10;
        System.out.println(class);
    }
}
```

Error:-

---------> The 'class' is pre defined keyword in java. Hence it cannot be used as a variable.

Corrected Code Snippet:-

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

Snippet 10:

```
public class Main {
    public void display() {
        System.out.println("No parameters");
    }
    public void display(int num) {
        System.out.println("With parameter: " + num);
    }
    public static void main(String[] args) {
        display();
        display(5);
    }
}
```

Error:-

---------> The method 'display' is not static and hence cannot be directly executed. To execute it we need to call it.

Corrected Code Snippet:-

```
public class Main {
    public void display() {
        System.out.println("No parameters");
    }

    public void display(int num) {
        System.out.println("With parameter: " + num);
    }
    public static void main(String[] args) {

        Main obj = new Main();


        obj.display();
        obj.display(5);
    }
}
```

Snippet 11:

```
public class Main {
    public static void main(String[] args) {
        int[] arr = {1, 2, 3};
        System.out.println(arr[5]);
    }
}
```

Error:-

---------> The array declared has only 3 elements and hence will give error due to array index being out of bound.

Corrected Code Snippet:-

```
public class Main {
    public static void main(String[] args) {
        int[] arr = {1, 2, 3};



        for (int i = 0; i < arr.length; i++) {
            System.out.println("Element at index " + i + ": " + arr[i]);
        }



        try {
            System.out.println(arr[5]);
        } catch (ArrayIndexOutOfBoundsException e) {
            System.out.println("Index out of bounds: " +
e.getMessage());
        }
    }
}
```

Snippet 12:

```
public class Main {
    public static void main(String[] args) {
        while (true) {
            System.out.println("Infinite Loop");
        }
    }
}
```

Error:-

---------> The loop will keep executing infinitely as there is no stopping condition.

Corrected Code Snippet:-

```
public class Main {
    public static void main(String[] args) {
        int count = 0;
        while (count < 5) {
            System.out.println("Loop iteration " + count);
            count++;
        }
    }
}
```

Snippet 13:

```
public class Main {
    public static void main(String[] args) {
        String str = null;
        System.out.println(str.length());
    }
}
```

Error:-

---------> The String is null and hence does not have any length value. Length does not work for null variables.

Corrected Code Snippet:-

```
public class Main {
    public static void main(String[] args) {
        String str = null;

        if (str != null) {
            System.out.println(str.length());
        } else {
            System.out.println("String is null.");
        }
    }
}
```

Snippet 14:

```
public class Main {
    public static void main(String[] args) {
        double num = "Hello";
        System.out.println(num);
    }
}
```

Error:-

---------> The double datatype should have decimal values.

Corrected Code Snippet:-

```
public class Main {
    public static void main(String[] args) {
        double num = 3.14;
        System.out.println(num);
    }
}
```

Snippet 15:

```
public static void main(String[] args) {
        int num1 = 10;
        double num2 = 5.5;
        int result = num1 + num2;
        System.out.println(result);
    }
}
```

Error:-

---------> The different datatype variables cannot be added. Hence we need to change datatype and make it same in order to add them or do arithmetic operations.

Corrected Code Snippet:-

```
public class Main {
    public static void main(String[] args) {
        int num1 = 10;
        double num2 = 5.5;
        int result = num1 + (int) num2;
        System.out.println(result);
    }
}
```

Snippet 16:

```
public class Main {
    public static void main(String[] args) {
        int num = 10;
        double result = num / 4;
        System.out.println(result);
    }
}
```

Error:-

---------> The datatype for output variable should be same as input.

Corrected Code Snippet:-

```
public class Main {
    public static void main(String[] args) {
        int num = 10;
        double result = (double) num / 4;
        System.out.println(result); // This will print 2.5
    }
}
```

Snippet 17:

```
public static void main(String[] args) {
        int a = 10;
        int b = 5;
        int result = a ** b;
        System.out.println(result);
    }
}
```

Error:-

---------> The power operation does not work in that manner in java.

Corrected Code Snippet:-

```
public class Main {
    public static void main(String[] args) {
        int a = 10;
        int b = 5;
        double result = Math.pow(a, b);
        System.out.println(result);
    }
}
```

Snippet 18:

```
public class Main {
    public static void main(String[] args) {
        int a = 10;
        int b = 5;
        int result = a + b * 2;
        System.out.println(result);
    }
}
```

Error:-

---------> The multiplication works first followed by addition.

Output:-

```
int result = a + b * 2;
5*2 = 10
10+10 = 20
```

Snippet 19:

```
public class Main {
    public static void main(String[] args) {
        int a = 10;
        int b = 0;
        int result = a / b;
        System.out.println(result);
    }
}
```

Error:-

---------> The number cannot be divide by zero. It will show arithmetic exception.

Corrected Code Snippet:-

```
public class Main {
    public static void main(String[] args) {
        int a = 10;
        int b = 0;

        if (b != 0) {
            int result = a / b;
            System.out.println(result);
        } else {
            System.out.println("Error: Division by zero is not
allowed.");
        }
    }
}
```

Snippet 20:

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

Error:-

---------> The print statement must end with a semicolon.

Corrected Code Snippet:-

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

Snippet 21:

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

Error:-

---------> The compiler gives syntax error.

Corrected Code Snippet:-

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

Snippet 22:

```
public static void main(String[] args) {
        static void displayMessage() {
            System.out.println("Message");
        }
    }
}
```

Error:-

---------> The 'display' method should be declared outside main method and for execution should be called within main nmethod.

Corrected Code Snippet:-

```
public class Main {

    public static void displayMessage() {
        System.out.println("Message");
    }

    public static void main(String[] args) {

        displayMessage();
    }
}
```

Snippet 23:

```
public class Confusion {
    public static void main(String[] args) {
        int value = 2;
        switch(value) {
            case 1:
                System.out.println("Value is 1");
            case 2:
                System.out.println("Value is 2");
            case 3:
                System.out.println("Value is 3");
            default:
                System.out.println("Default case");
        }
    }
```

Error:-

---------> The default statement executes even after condition is false. To exclude executing it remove default case.

Snippet 24:

```
public class Switch {
    public static void main(String[] args) {
        double score = 85.0;
        switch(score) {
            case 100:
                System.out.println("Perfect score!");
                break;
            case 85:
                System.out.println("Great job!");
                break;
            default:
                System.out.println("Keep trying!");
        }
    }
}
```

Error:-

---------> To use the switch statement with score, you should either change score to an int (or another supported type) or use an if-else statement instead.

Corrected Code Snippet:-

```
public class Switch {
    public static void main(String[] args) {
        double score = 85.0;
        if (score == 100.0) {
            System.out.println("Perfect score!");
        } else if (score == 85.0) {
            System.out.println("Great job!");
        } else {
            System.out.println("Keep trying!");
        }
    }
}
```

Snippet 25:

```
public class Switch {
    public static void main(String[] args) {
        int number = 5;
        switch(number) {
            case 5:
                System.out.println("Number is 5");



                break;
            case 5:
                System.out.println("This is another case 5");
                break;
            default:
                System.out.println("This is the default case");
        }
    }
}
```

Error:-

---------> Duplicate case Labels: The case 5: label appears twice. This is not allowed because each case label must be unique within a switch statement.

Corrected Code Snippet:-

```
public class Switch {
    public static void main(String[] args) {
        int number = 5;
        switch(number) {
            case 5:
                System.out.println("Number is 5");
                // Additional code for case 5 can be added here
                break;
            default:
                System.out.println("This is the default case");
        }
    }
}
```

Snippet 26:

```
public class MissingBreakCase {
    public static void main(String[] args) {
        int level = 1;
        switch(level) {
            case 1:
                System.out.println("Level 1");
            case 2:
                System.out.println("Level 2");
            case 3:
                System.out.println("Level 3");
            default:
                System.out.println("Unknown level");
        }
    }
}
```

Error:-

---------> The break statement will stop execution whenever condition is satisfied.
