Section 1

Snippet 1:

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

Error: Main method not found in class Main, please define the main method as: public static void main(String[] args)
Here static is missing.

Corrected:

```
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: Main method not found in class Main, please define the main method as: public static void main(String[] args)
Here public is missing

Corrected:

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

Snippet 3:

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

Error: Main method not found in class Main, please define the main method as:
  public static void main(String[] args)

Corrected:

```
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: Main method not found in class Main, please define the main method as: public static void main(String[] args)

Corrected:

```
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: No error

Output: Main method with String[] args

Corrected:

Snippet 6:

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

Error: Compile time error: cannot find symbol (variable y is not declared).

Corrected:

```
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: Compile time error: incompatible types: String cannot be converted to int.

Corrected:

```
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: error: ')' or ',' expected

Corrected:

```
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: Compile time error : not a statement
Corrected:

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

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: Compile time error: non-static method display() cannot be referenced from a static context
Corrected:

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

Snippet 11:

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

Error: Compile Time Error: Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: Index 5 out of bounds for length 3

Corrected:

```
public class Main {
  public static void main(String[] args) {
    int[] arr = {1, 2, 3};
    System.out.println(arr[0]);
  }
}
```

Snippet 12:

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

Error: There will be Infinite Loop

Corrected:

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

Snippet 13:

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

Error: Run Time Error: Exception in thread "main" java.lang.NullPointerException: Cannot invoke "String.length()" because "<local1>" is null

Corrected:

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

Snippet 14:

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

```
        System.out.println(num);
    }
}
```

Error: Compile Time Error: incompatible types: String cannot be converted to double

Corrected:

```
public class Main {
    public static void main(String[] args) {
        double num = 9.85;
        System.out.println(num);
    }
}
```

Snippet 15:

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

Error: Compile Time Error: incompatible types: possible lossy conversion from double to int

Corrected:

```
public class Main {
    public static void main(String[] args) {
        int num1 = 10;
        double num2 = 5.5;
                             int num3 = (int)num2;
        int result = num1 + num3;
        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: Expected result was 2.5 but output was 2

Corrected:

```
public class Main {
    public static void main(String[] args) {
        int num = 10;
        double result = num / 4.0;
        System.out.println(result);
    }
}
```

Snippet 17:

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

Error: Compile Time Error: illegal start of expression

Corrected:

```
public class Main {
    public static void main(String[] args) {
        int a = 10;
        int b = 5;
        int result = 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);
    }
}
```

Output: 20

Here precedence of (*) is higher than (+)

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: Exception in thread "main" java.lang.ArithmeticException: / by zero

In Java, dividing an integer by zero is undefined because mathematically, division by zero does not result in a finite number.

Snippet 20:

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

Error: ';' expected

Corrected:

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

Snippet 21:

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

Error: reached end of file while parsing }

Corrected:

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

Snippet 22:

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

Error: Compile Time Error: illegal start of expression, class, interface, enum, or record expected }
A method cannot be declared inside another method. Methods must be declared within a class.

Corrected:

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

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");
        }
    }
}
```

In Java, once a matching case is found and executed, the program will "fall through" and execute all subsequent cases until a ‘break’ statement or the end of the switch block is reached.
To avoid this, we need to add ‘break’ statements at the end of each case.

Corrected:

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

Snippet 24:

```
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");
        }
    }
}
```

The program will "fall through" and execute all subsequent cases until a ‘break’ statement or the end of the switch block is reached.
To avoid this, we need to add ‘break’ statements at the end of each case.

Corrected:

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

Snippet 25:

```
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: Compile Time Error: selector type double is not allowed.

Corrected:

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

```
        }
    }
}
```

Snippet 26:

```
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: Compile Time Error: duplicate case label.

Duplicate case label is not valid in Java.

Corrected:

```
class Switch {
    public static void main(String[] args) {
        int number = 5;
        switch(number) {
            case 1:
                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");
        }
    }
}
```