Snippet 1

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

Ans :- Main method is not static in class Main

Correct Code : -

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

Ans :- Main method not found in class Main

Correct Code : -

```
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;
        }
}
```

Ans : - Main method must return a value of type void in class Main.
As this method is used to start execution of the code and it doesn’t return any value .

Correct Code:-

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

Ans : - Main method not found in class Main. The String [] args parameter in the `main` method allows Java programs to accept command-line arguments when they are executed.

Correct Code: -

```
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");

} }
```

Ans : - There Should be only one main method in the code with String [] args. Hence in this code the second main method with int [] args will not run.

Correct Code : -

```
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");

} }
```

————————————————————————————————————-

Snippet 6

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

Ans :- Here the letter Y is not declared as Variable and used In code .

Correct Code:-

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

        }

}
```

Ans : - Here “Hello” is the String. But the variable x is of Integer type. If you want to use this then we need to do explicit casting .

Correct Code:-

```
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!" }
}
```

Ans : - Here is the problem in SOP statement . The brackets are not closed after the String .Due to this the compilation of code could not go forward.

Correct Code:-

```
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); } }
```

Ans : - Here the error is variable class . We can not declare Java Keywords as a Variable.

Correct Code:-

```
public class Main {

        public static void main(String[] args) {

                int class1 = 10;

                System.out.println(class1); }

}
```

———————————————————————————————————-

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); }
}
```

Ans : - Here the Non static methods Display is called in the Static main method which is not allowed. So we have to make both the methods as static.

Correct Code : -

```
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]); }
}
```

Ans : - Here the total Array elements are 3 means the arr length is 2 (0,1,2). Here SOP statement is accessing the value of arr[5] hence we get error “Index 5 out of bound”.

Correct Code : -

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

——————————————————————————————————————-

Snippet 12

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

Ans : - Here this code will run infinite times because the condition is true.we have to modify the condition.

Correct Code : -

```
 public class Main {
        public static void main(String[] args) {
                int i = 0;
                while (i < 6) {
                        System.out.println(i);
                } }
}
```

——————————————————————————————————————-

Snippet 13

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

Ans : -Here str is declared as String but is assigned null, meaning it doesn't point to any actual object in memory. And str,length() will try to get object from the memory and will throw null point exception.

Correct Code : -

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

——————————————————————————————————————-

Snippet 14

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

Ans:-Here String cannot be converted in double . We have to make variable num of String literal.

Correct Code : -

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

Ans : - Here num1 is of int type and num2 is of double type so java convert num1 to double. So Double + Double = Double . So we have to make result variable of type Double.

Correct Code:-

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

Ans :- The code is correct I am looking for same output.

——————————————————————————————————————-

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);

} }
```

Ans : - Java doesn’t follow the ** operator. It is used in python to find power of a to b.In java we can use Math.pow(a,b). And Math.pow() will give double type value.

Correct Code:-

```
import java.util.*;

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);

} }
```

Ans : - The Code output is 20. First b is multiplied by 2 and then added with a. And the stored in result .

——————————————————————————————————————-

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);
} }
```

Ans : - Java can not divide any number by 0 because java doesn’t know what integer to be return because there is no valid outcome.

Correct Code:-

```
public class Main {
        public static void main(String[] args) {
                int a = 10;
                int b = 1; //must be non zero value
                int result = a / b;
                System.out.println(result);
} }
```

————————————————————————————————————-

Snippet 20

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

Ans : - SOP statement must be end with semicolon(;) Because Java compiler do not know where to end the execution of statement.

Correct Code : -

```
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

}
```

Ans : -Error : Reached end of file while parsing . There should be closing braces for every opening braces.

Correct Code : -

```
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"); } }
```

Ans : -Java does not allow you to define a method inside another method.If we want to make the static method then it should be declared outside the main method.

Correct Code : -

```
public class Main {

     public static void main(String[] args) {

          displayMessage();  // Calling the static method

     }

     public 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");

} }

}
```

Ans : - Here the we will get the output as (2,3,Default case) if we want exact answer then we should use break statement after every case.

Correct Code :-

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

Ans : - It prints all value because the break statement is missing if we use break statement then it will print only exact value.

Correct Code :

```
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!");

} }

}
```

Ans : - Switch statement can only supports Integral type like Int , short, char, long).so if we want to run this program then we have to typecast the double to int

Correct Code:-

```
public class Switch {
    public static void main(String[] args) {

        double score = 85.0;
        switch((int)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"); }

} }
```

Ans : - Here the case labels are have same value which is not allowed in java. To solve this issue we have to make the label name unique.

Correct Code:-

```
public class Switch {
        public static void main(String[] args) {

                int number = 5;
```

```
    switch(number) {

        case 5:
            System.out.println("Number is 5");

            break;

        case 4:

            System.out.println("This is another case 5");

            break;

        default:

            System.out.println("This is the default case"); }

} }
```
