/* 1. ERROR: Main method is not static in class Main.

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

/* 2. ERROR:Compilation gets succesful, but at the time of execution throws an error: Main method not found in class Main, please define the main method as: public static void main(String[] args)

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

/* 3. ERROR: Main method must return a value of type void in class Main, please define the main method as: public static void main(String[] args), Then meaning of void is null, in java void keyword is used in main method to indicate that the method does not return any value.

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

/*4. ERROR:Compilation gets succesful, but at the time of execution throws an error: Main method not found in class Main, please define the main method: public static void main(String[] args),String args[] is used to pass command-line arguments to a Java program.

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

/*5. Yes, you can have multiple main methods in java but having different in parameters.

```
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");
 }
}
*/
```

/* 6. ERROR: cannot find symbol int x = y + 10; variable is a name given to a memory location. It is the basic unit of storage in a program.

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

/* 7. ERROR: incompatible types: String cannot be converted to int, Java enforces type safety to ensure that code is reliable, secure, and predictable.

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

```
*/
```

/* 8.ERROR:Closing parenthesis is not there that's why at the time of compilation it throws an error

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

/* 9. ERROR: not a statement int class = 10; Reserved keywords cannot be used as identifiers because they have predefined meanings in the language's syntax

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

/* 10. ERROR: non-static method display() cannot be referenced from a static context, Yes, method overloading is allowed in Java.

display();

```
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);
 }
}
*/
```

/* 11. ERROR: illegal character: '\u201a', because it tries to access an index (5) that doesn't exist in the array, which only has indices 0, 1, and 2.

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

/* 12. It will print the statement in infinite times till the condition is true, by using conditional statements we can avoid

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

/* 13. ERROR: Exception in thread "main" java.lang.NullPointerException: Cannot invoke "String.length()" ,The exception occurs because str is null, and calling a method (length()) on a null reference is not allowed.

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

/* 14. ERROR: incompatible types: String cannot be converted to double., Java enforces data type constraints because "Hello" is a String, but num is declared as a double

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

/* 15. ERROR: incompatible types: possible lossy conversion from double to int, we cand handle this by using explicit type casting

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

/* 16. OUTPUT: 2.0 , EXPECTED: 2.0

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

/* 17. ERROR: error: illegal start of expression, because Java uses Math.pow() for exponentiation.

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

/* 18. OUTPUT: 20, Operators with higher precedence are evaluated before those with lower precedence.

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

/* 19. ERROR: ArithmeticException: / by zero at Main.main(Main.java:182), Division by zero is not allowed because it is mathematically undefined. In Java, dividing by zero in integer operations causes this exception

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

```
/* 20. ERROR ; Expected, it will cause compilation error
public class Main {
 public static void main(String[] args) {
 System.out.println("Hello, World")
 }
}
*/
```

```
/* 21. ERROR: reached end of file while parsing },
public class Main {
 public static void main(String[] args) {
 System.out.println("Hello, World!");
 // Missing closing brace here
}
*/
```

```
/* 22. ERROR:  illegal start of expression,No, a method cannot be declared inside another method in Java. Methods must be declared directly within a class, not inside other methods.
public class Main {
 public static void main(String[] args) {
 static void displayMessage() {
 System.out.println("Message");
 }
 }
}
*/
```

```
/* 23. by using break statement we can remove the default case.
public class Main {
 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");
 }
 }
}
*/
```

```
/* 24. Break is used terminate the processing of a particular case within a switch statement.
public class Main{
 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");
 }
 }
}
*/
```

/* 25. ERROR: switch expression must be of type byte, short, int, char, or an enum type.

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

/* 26. ERROR: duplicate case label.

```
public class Switch {
    public static void main(String[] args) {
        int number = 5;
        switch(number) {
            case 5:
                System.out.println("Number is 5");
                break;
            // Grouping multiple cases that should execute the same code
            // case 6:
            // case 7:
            //    System.out.println("Another case");
            //    break;
            default:
              System.out.println("This is the default case");
        }
    }
}
```

Q1. Write a program to classify student grades based on the following criteria:

- **If the score is greater than or equal to 90, print "A"**
- **If the score is between 80 and 89, print "B"**
- **If the score is between 70 and 79, print "C"**
- **If the score is between 60 and 69, print "D"**
- **If the score is less than 60, print "F"**

```
class Statements1 {
public static void main(String args[]){
        int marks = 95;
        if (marks >= 90){
                System.out.println(" A");
        }else if(marks >=80 && marks <= 89){
                System.out.println("Grade B");
        }
        else if(marks >=70 && marks <=79){
                System.out.println("Grade C");
        }
        else if(marks >=60& marks <=69){
                System.out.println("Grade D");
        }
        else if(marks <60){
                System.out.println("Grade F");
        }else{
                System.out.println("FAIL");
        }
}
}
```

Question 2: Days of the Week Write a program that uses a nested switch statement to print out the day of the week based on an integer input (1 for Monday, 2 for Tuesday, etc.). Additionally, within each day, print whether it is a weekday or weekend.

```
public class DaysOfWeek {
  public static void main(String[] args) {
    int day = 4;

    switch (day) {
      case 1: case 2: case 3: case 4: case 5:
        switch (day) {
          case 1: System.out.println("Monday - Weekday"); break;
          case 2: System.out.println("Tuesday - Weekday"); break;
          case 3: System.out.println("Wednesday - Weekday"); break;
          case 4: System.out.println("Thursday - Weekday"); break;
          case 5: System.out.println("Friday - Weekday"); break;
        }
        break;
      case 6: case 7:
        switch (day) {
          case 6: System.out.println("Saturday - Weekend"); break;
          case 7: System.out.println("Sunday - Weekend"); break;
        }
        break;
      default:
        System.out.println("Invalid day");
    }
  }
}
```

Question 3: Calculator Write a program that acts as a simple calculator. It should accept two numbers and an operator (+, -, *, /) as input. Use a switch statement to perform the appropriate operation. Use nested ifelse to check if division by zero is attempted and display an error message.

```
class Statements2 {

public static void main(String args[]){

int a = 6;

int b = 5;

char operator = '/';

int result;

switch(operator){

   case 1:

    result= a + b;

         System.out.println("Result:" +result);

         break;

        case 2:

    result= a - b;

         System.out.println("Result:" +result);

         break;

        case 3:

    result= a * b;

         System.out.println("Result:" +result);

         break;

        case 4:

    if (b != 0) {

           result = a / b;

           System.out.println("Result: " + result);

         } else {

           System.out.println("Error: Division by zero is not allowed.");

         }

          break;

   default:
    System.out.println("Error: Invalid operator. Please use +, -, *, or /.");
    break;
        }
}
}
```

Question 4: Discount Calculation Write a program to calculate the discount based on the total purchase amount. Use the following criteria:

- **If the total purchase is greater than or equal to Rs.1000, apply a 20% discount.**
- **If the total purchase is between Rs.500 and Rs.999, apply a 10% discount.**
- **If the total purchase is less than Rs.500, apply a 5% discount.**

Additionally, if the user has a membership card, increase the discount by 5%.

```
class DiscountCal{
        public static void main(String[] args){
                int amount = 100;
                double dis = 0;

                if(amount >= 1000){
                        dis = amount*0.2;
                        System.out.println("The price is "+ (amount - dis));
                } else if(amount > 500 && amount <= 999){
                        dis = amount*0.1;
                        System.out.println("The price is "+ (amount - dis));
                } else {
                        dis = amount*0.05;
                        System.out.println("The price is "+ (amount - dis));
                }
        }
}
```

Question 5: Student Pass/Fail Status with Nested Switch Write a program that determines whether a student passes or fails based on their grades in three subjects. If the student scores more than 40 in all subjects, they pass. If the student fails in one or more subjects, print the number of subjects they failed in.

```
class PassFail{
    public static void main(String[] args){
        int sub1 = 10;
        int sub2 = 40;
        int sub3 = 20;
        int pass = 0;
        int fail = 0;

        if (sub1 >= 40){
            pass++;
        } else {
            fail++;
        }
        if (sub2 >= 40){
            pass++;
        } else {
            fail++;
        }
        if (sub3 >= 40){
            pass++;
        } else {
            fail++;
        }


        if(pass==3){
            System.out.println("Passed");
        }else{
            switch(fail){
```

```
                        case 1:System.out.println("Failed in 1 subject.");
                        break;
                        case 2:System.out.println("Failed in 2 subject.");
                        break;
                        case 3:System.out.println("Failed in 3 subject.");
                        break;

            }

        }

    }
    }
```