

# ASSIGNMENT 2

## Snippet 1

What error do you get when running this code?

main.java:1: error: class Main is public, should be declared in a file named Main.java

Error is causing due to the missing static keyword in the snippet .

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

## Snippet 2

- What happens when you compile and run this code?

main.java:1: error: class Main is public, should be declared in a file named Main.java

```
public class Main {
```

Error is causing due to the missing of Public Keyword in the snippet .

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

## Snippet 3

What error do you encounter? Why is void used in the main method?

main.java:5: error: reached end of file while parsing

```
}
```

Error is causing due to the int keyword replaced with void for the main method it must be returning void .

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

# ASSIGNMENT 2

```
}
```

## Snippet 4

What happens when you compile and run this code? Why is String[] args needed?

main.java:1: error: class Main is public, should be declared in a file named Main.java

```
public class Main {
```

Error is causing due to the missing of String args [] which is the entry point parameter of code .

```
public class main {
```

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

```
        System.out.println("Hello, World!");
```

```
    }
```

```
}
```

## Snippet 5

- Can you have multiple main methods? What do you observe?

NO ERROR , Cause method overloading is allowed In java , however there will be only one main method for the entry point of the code .

Main method with String[] args

```
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");
```

```
    }
```

```
}
```

# ASSIGNMENT 2

## Snippet 6

- What error occurs? Why must variables be declared?

main.java:3: error: cannot find symbol

```
int x = y + 10;
```

^

symbol: variable y

location: class main

Error is causing due to not declaring of y variable , I declared as 5 then we can get the output as 15

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

## Snippet 7

- What compilation error do you see? Why does Java enforce type safety?

main.java:3: error: incompatible types: String cannot be converted to int

```
int x = "Hello";
```

Error is causing due to the assigning int as string data type

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

# ASSIGNMENT 2

## Snippet 8

- What syntax errors are present? How do they affect compilation?

main.java:3: error: ')' or ';' expected

```
System.out.println("Hello, World!"
```

^

Error is causing due to the missing parenthesis and semicolon.

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

## Snippet 9

- What error occurs? Why can't reserved keywords be used as identifiers?

main.java:3: error: not a statement

```
int class = 10;
```

^

main.java:3: error: ';' expected

```
int class = 10;
```

^

main.java:3: error: <identifier> expected

```
int class = 10;
```

^

main.java:4: error: illegal start of expression

```
System.out.println(class);
```

^

main.java:4: error: <identifier> expected

```
System.out.println(class);
```

Error is causing due to the class keyword which is a reserved keyword

# ASSIGNMENT 2

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

## Snippet 10

- What happens when you compile and run this code? Is method overloading allowed?

main.java:9: error: non-static method display() cannot be referenced from a static context

```
    display();  
    ^
```

main.java:10: error: non-static method display(int) cannot be referenced from a static context

```
    display(5);
```

Error is causing due to the display() , so for that we need to create new main instance to create display() method .

## Snippet 11

- What runtime exception do you encounter? Why does it occur?

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

```
    at main.main(main.java:4)
```

Error is causing due to the array indexing of assign the values which is not in the array and not even the length .

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

# ASSIGNMENT 2

## Snippet 12

- What happens when you run this code? How can you avoid infinite loops?

main.java:1: error: class Main is public, should be declared in a file named Main.java

```
public class Main {  
    ^
```

NO ERROR but it is going in the infinite loop due to the missing keyword of break .

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

## Snippet 13

- What exception is thrown? Why does it occur?

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

at main.main(main.java:4)

Error is causing due to the null in front of string data type which can't be possible

And it will give output as nullpointer cause it is also a keyword

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

# ASSIGNMENT 2

## Snippet 14

- What compilation error occurs? Why does Java enforce data type constraints?

main.java:3: error: incompatible types: String cannot be converted to double

```
double num = "Hello";
```

^

Error was causing due to the hello written in the double data type

```
public class main {  
    public static void main(String[] args) {  
        double num = 10.5;  
        System.out.println(num);  
    }  
}
```

## Snippet 15

What error occurs when compiling this code? How should you handle different data types in operations?

main.java:5: error: incompatible types: possible lossy conversion from double to int

```
int result = num1 + num2;
```

Error was causing due to the int datatype which is using for result variable but we are getting the decimal values so we will need to use double data type instead of Int .

```
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

- What is the result of this operation? Is the output what you expected?

# ASSIGNMENT 2

Error is not there but also we are not expecting the value as we are getting due to the int datatype for that we need to just type as 4.0 to get the output as 2.5

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

## Snippet 17

- What compilation error occurs? Why is the \*\* operator not valid in Java?

main.java:5: error: illegal start of expression

```
int result = a ** b;  
            ^
```

There is not a error but Java Doesn't support the '\*\*' operator .

```
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

- What is the output of this code? How does operator precedence affect the result?

There is no error as in java multiplication happens before the addition in java .



# ASSIGNMENT 2

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

## Snippet 19

- What runtime exception is thrown? Why does division by zero cause an issue in Java?

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

at main.main(main.java:5)

IT is not able to divide by zero

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

## Snippet 20

- What syntax error occurs? How does the missing semicolon affect compilation?

main.java:3: error: ';' expected

```
System.out.println("Hello, World")
```

^

Semicolon was missing after the statement .

# ASSIGNMENT 2

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

## Snippet 21

- What does the compiler say about mismatched braces?

main.java:4: error: reached end of file while parsing

```
    }  
    ^
```

Missing Bracket

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

## Snippet 22

- What syntax error occurs? Can a method be declared inside another method?

main.java:3: error: illegal start of expression

```
    static void displayMessage() {  
    ^
```

main.java:7: error: class, interface, enum, or record expected

```
    }  
    ^
```

# ASSIGNMENT 2

## Snippet 23

Error to Investigate: Why does the default case print after "Value is 2"? How can you prevent the program from executing the default case?

Value is 2

Value is 3

Default case

NO ERROR here just the expected output was different than getting . due to the missing break keyword .

```
public class main {  
    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");  
                break;  
        }  
    }  
}
```

# ASSIGNMENT 2

## Snippet 24

- Error to Investigate: When level is 1, why does it print "Level 1", "Level 2", "Level 3", and "Unknown level"? What is the role of the break statement in this situation?

Level 1

Level 2

Level 3

Unknown level

NO ERROR just break keyword was missing from the code .

```
public class main {  
    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");  
                break;  
        }  
    }  
}
```

# ASSIGNMENT 2

## Snippet 25

- Error to Investigate: Why does this code not compile? What does the error tell you about the types allowed in switch expressions? How can you modify the code to make it work?

main.java:4: error: patterns in switch statements are a preview feature and are disabled by default.

```
switch(score) {  
    ^
```

(use --enable-preview to enable patterns in switch statements)

main.java:5: error: constant label of type int is not compatible with switch selector type double

```
case 100:  
    ^
```

main.java:8: error: constant label of type int is not compatible with switch selector type double

```
case 85:
```

Switch statements in java supports certain types int , char , byte , short , string it doesn't support double data type

```
public class main {  
    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!");  
        }  
    }  
}
```

# ASSIGNMENT 2

## Snippet 26

- Error to Investigate: Why does the compiler complain about duplicate case labels? What happens when you have two identical case labels in the same switch block?

main.java:8: error: duplicate case label

```
    case 5:
```

```
    ^
```

Error was due to the duplicate assign cases in the code .

```
public class main {  
    public static void main(String[] args) {  
        int number = 5;  
        switch(number) {  
            case 5:  
                System.out.println("Number is 5");  
                break;  
            case 6:  
                System.out.println("This is another case 5");  
                break;  
            default:  
                System.out.println("This is the default case");  
        }  
    }  
}
```

# ASSIGNMENT 2

## SECTION 2

Question 1: Grade Classification 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"

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

# ASSIGNMENT 2

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 main {  
    public static void main (String args[]){  
        int day = 4 ;  
        switch (day){  
            case 1 : System.out.println("Monday");  
            break;  
  
            case 2 : System.out.println("Tuesday");  
            break;  
  
            case 3 : System.out.println("Wednesday");  
            break;  
  
            case 4 : System.out.println("Thursday");  
            break;  
  
            case 5 : System.out.println("Friday");  
            break;  
  
            case 6 : System.out.println("Saturday");  
            break;  
  
            case 7 : System.out.println("Sunday");  
            break;  
  
            default : System.out.println("Invalid Input");  
            break;  
        }  
    }  
}
```



# ASSIGNMENT 2

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 if else to check if division by zero is attempted and display an error message.

```
public class main {  
    public static void main (String args[]){  
        int value1 = 4 ;  
        int value2 = 4 ;  
        char operator = '%';  
        switch (operator){  
            case '+' : System.out.println("Result is "+(value1+value2));  
            break;  
  
            case '/' : System.out.println("Result is "+(value1/value2));  
            break;  
  
            case '-' : System.out.println("Result is "+(value1-value2));  
            break;  
  
            case '*': System.out.println("Result is "+(value1*value2));  
            break;  
  
            case '%': System.out.println("Result is "+(value1%value2));  
            break;  
  
            default : System.out.println("Invalid Input");  
            break;  
        }  
    }  
}
```

# ASSIGNMENT 2

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%.

# ASSIGNMENT 2

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.

```
public class main {  
    public static void main (String args[]){  
  
        int subject = 38 ;  
        int subject2 = 20 ;  
        int subject3 = 50 , failcount = 0 ;  
  
        if (subject <= 40 ) failcount++ ;  
        if (subject2 <= 40 ) failcount++ ;  
        if ( subject3 <=40 ) failcount++ ;  
  
        if (subject >= 40 && subject2 >= 40 && subject3 >= 40 ) {  
  
            System.out.println("Student is Passed " ) ;  
        }  
        else {  
            System.out.println("Student is failed in " + failcount + " Subject ");  
        }  
  
    }  
}
```