Lab Assignment 2

Snippet 1:

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

Main method is not static please define static main method.

Snippet 2:

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

Main method not found in class main

Snippet 3:

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

It reached end of file while parsing because one closing curly bracket is missing. Main method must written value of type void in class main. Unexpected return type.

We use void because void means our method does not return any value. If method does not have void then it must return something(int, String, etc.). Hence **void** is must.

Snippet 4:

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

Main method does not found in class main.

“String args[]” is used to passed command line argument to Java program. It allows external data to be passed into program when into program when it starts.

Snippet 5:

```
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");
    }
}
```

If we have multiple main methods in our program then only first main method get executed others didn’t.

In above code only highlighted main method got executed.

Snippet 6:

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

```
    {
        int y = 0;
        int x = y + 10;
        System.out.println(x);
    }
}
```

Cannot find symbol y. And symbol must be initialized or declared. Otherwise it will give error.

Snippet 7:

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

String cannot be converted into string.

Type safety ensure that any object, method should manipulate is of proper type.

Snippet 8:

```
class Main
```

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

‘)’ and ‘;’ expected. Otherwise it won’t compile a code.

Snippet 9:

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

“int class = 10” is not a statement and also got error of illegal start of expression. “int class = 10” instead of class identifier expected.

We can't used reserved keywords as identifiers because keywords have no user defined meaning.

Snippet 10:

```
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);
	}
}
```

Non static method display() & display(5) cannot be reference from a static contex.

Yes, method overloading is allowed.

Snippet 11:

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

```
    {
        int[] arr = {1, 2, 3};
        System.out.println(arr[2]);
    }
}
```

Got error like ArrayIndexOutOfBound because Index 5 is out of bound for array length 3.

Snippet 12:

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

It was giving infinite loop because there was not breaking in the loop.

So to break that infinite loop we have to use break keyword in the while loop.

Snippet 13:

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

It throws NullPointerException error because string str(<local>) is null.

Snippet 14:

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

What compilation error occurs?

String cannot be converted to double hence we have to use string instead of double.

Why does Java enforce data type constraints?

It is typically used to improve code readability & catch type related errors at compiled time.

Snippet 15:

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

What error occurs when compiling this code? -> Incompatible type

How should you handle different data types in operations?

Use data type of higher byte amoung them because data type with lower byte can easily get converted to data type with higher byte.

Snippet 16:

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

```
    {
        double num = 10;
        double result = num / 4;
        System.out.println(result);
    }
}
```

What is the result of this operation? 2.0

Is the output what you expected? Use double instead of int then only we got expected output. That is 2.5.

Snippet 17:

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

What compilation error occurs? Illegale start of expression

Why is the ** operator not valid in Java? It’s not valid multiplication operator.

Snippet 18:

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

What is the output of this code? -> 20

How does operator precedence affect the result?

Operator with precedence evaluated first than operator with low precedence.

Snippet 19:

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

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

What runtime exception is thrown? ArithmaticException

Why does division by zero cause an issue in Java?

Dividing integer by zero will result in exception. Values like infinite are for floating point not for integer. So, use float instead of int.

Snippet 20:

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

What syntax error occurs? ‘;’ expected

How does the missing semicolon affect compilation? Generate compilation error.

Snippet 21:

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

```
    {
        System.out.println("Hello, World!");
        // Missing closing brace here
    }
}
```

What does the compiler say about mismatched braces?

It reached at the end of file while parsing.

Snippet 22:

```
class Main
{
    static void displayMessage()
        {
        System.out.println("Message");
        }

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

What syntax error occurs? Illegale start of expression.

Can a method be declared inside another method?

Can’t declared a method directly into the method.

Snippet 23:

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

Why does the default case print after "Value is 2"? Because there is no break in between the case.

How can you prevent the program from executing the default case?

Using break we can prevent program from executing default case. But if value doesn't match with the case then default will execute.

Snippet 24:

```
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");
```

```
        }
    }
}
```

When level is 1, why does it print "Level 1", "Level 2", "Level 3", and "Unknown level"? Because there is no breaking in cases.

What is the role of the break statement in this situation?

Break stop execution for next case it only execute the expected case. If it is not available then it will print or execute default case.

Snippet 25:

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

- Why does this code not compile?

  Because switch only work with byte, short, char, int, string, enum types it doesn’t work with double.

- What does the error tell you about the types allowed in switch expressions?

  switch only work with byte, short, char, int, string, enum types

- How can you modify the code to make it work?

  We have to use if-else statement to make it work.

Snippet 26:

```
class Switch
{
    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");
            }
        }
}
```

Why does the compiler complain about duplicate case labels?

Because it has 2 ‘case 5’ so it gives duplicate case label error.

What happens when you have two identical case labels in the same switch block?

It gives duplicate case label error.

Question 1: Grade Classification

```
class Main
{
    public static void main(String args[])
    {
        int score = 60;
        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 if(score < 60)
            {
                System.out.println("F");
            }
            else
            {
                System.out.println("You are Failed");
            }
        }
}
```

Question 2: Days of the Week

```
class Main
{
	public static void main(String args[])
	{
		int num = 10;
		switch(num)
		{
			case 1:
				System.out.println("Monday");
				System.out.println("It's a WeekDay");
				break;
			case 2:
				System.out.println("Tuesday");
				System.out.println("It's a WeekDay");
				break;
			case 3:
				System.out.println("Wendsday");
				System.out.println("It's a WeekDay");
				break;
			case 4:
				System.out.println("Thursday");
				System.out.println("WeekDay");
				break;
```

```
                case 5:
                        System.out.println("Friday");
                        System.out.println("It's a WeekDay");
                        break;
                case 6:
                        System.out.println("Saturday");
                        System.out.println("It's a WeekEnd");
                        break;
                case 7:
                        System.out.println("Sunday");
                        System.out.println("It's a WeekEnd");
                        break;
                default:
                        System.out.println("Invalid Input");
            }
        }
}
```

Question 3: Calculator

```
import java.util.Scanner;
class Main
{
        public static void main(String args[])
        {
```

```
		Scanner sc = new Scanner(System.in);

		System.out.print("Enter a number 1 : ");
		int num1 = sc.nextInt();

		System.out.print("Enter a number 2 : ");
		int num2 = sc.nextInt();

		System.out.print("Enter an operator (+,-,*,/) : ");
		char operator = sc.next().charAt(0);

		int result;
		switch(operator)
		{
			case '+':
				result = num1 + num2;
				System.out.println("Addition of numbers is : "
+ result);
				break;
			case '-':
				result = num1 - num2;
				System.out.println("Substraction of numbers is
: " + result);
				break;
```

```
                case '*':
                        result = num1 * num2;
                        System.out.println("Multiplication of numbers
is : " + result);
                        break;
                case '/':
                        if(num2 == 0)
                        {
                                System.out.println("Division by zero is
not acceptable");
                        }
                        else
                        {
                                result = num1 / num2;
                                System.out.println("Division of numbers
is : " + result);
                        }
                        break;
                default:
                        System.out.println("Invalid Input!");
            }
            sc.close();
        }
}
```

Question 4: Discount Calculation

```
class Main
{
	public static void main(String args[])
	{
		int total = 200;
		boolean membershipCard = true;
		double discount = 0.0;
		if(total >= 1000)
		{
			discount = 20.0;
		}
		else if(total >= 500 && total <= 999)
		{
			discount = 10.0;
		}
		else if(total < 500)
		{
			discount = 5.0;
		}
		if(membershipCard)
		{
				discount = discount + 5.0;
		}
```

```
        double discountAmount = (discount*total)/100;
        System.out.println("Discount on total purchase is : " +
discountAmount);
    }
}
```

Question 5: Student Pass/Fail Status with Nested Switch

```
class Main
{
    public static void main(String args[])
    {
        int sub1 = 40;
        int sub2 = 10;
        int sub3 = 60;
        int count = 0;

        if(sub1<=40)
        {
            count++;
        }
        else if(sub2<=40)
        {
            count++;
        }
```

```
            else if(sub3<=40)
            {
                count++;
            }
            if(count > 0)
            {
                System.out.println("You failled the exam.");
            }
            else
            {
                System.out.println("Congratulation you have Passed
the exam.");
            }
        }
}
```