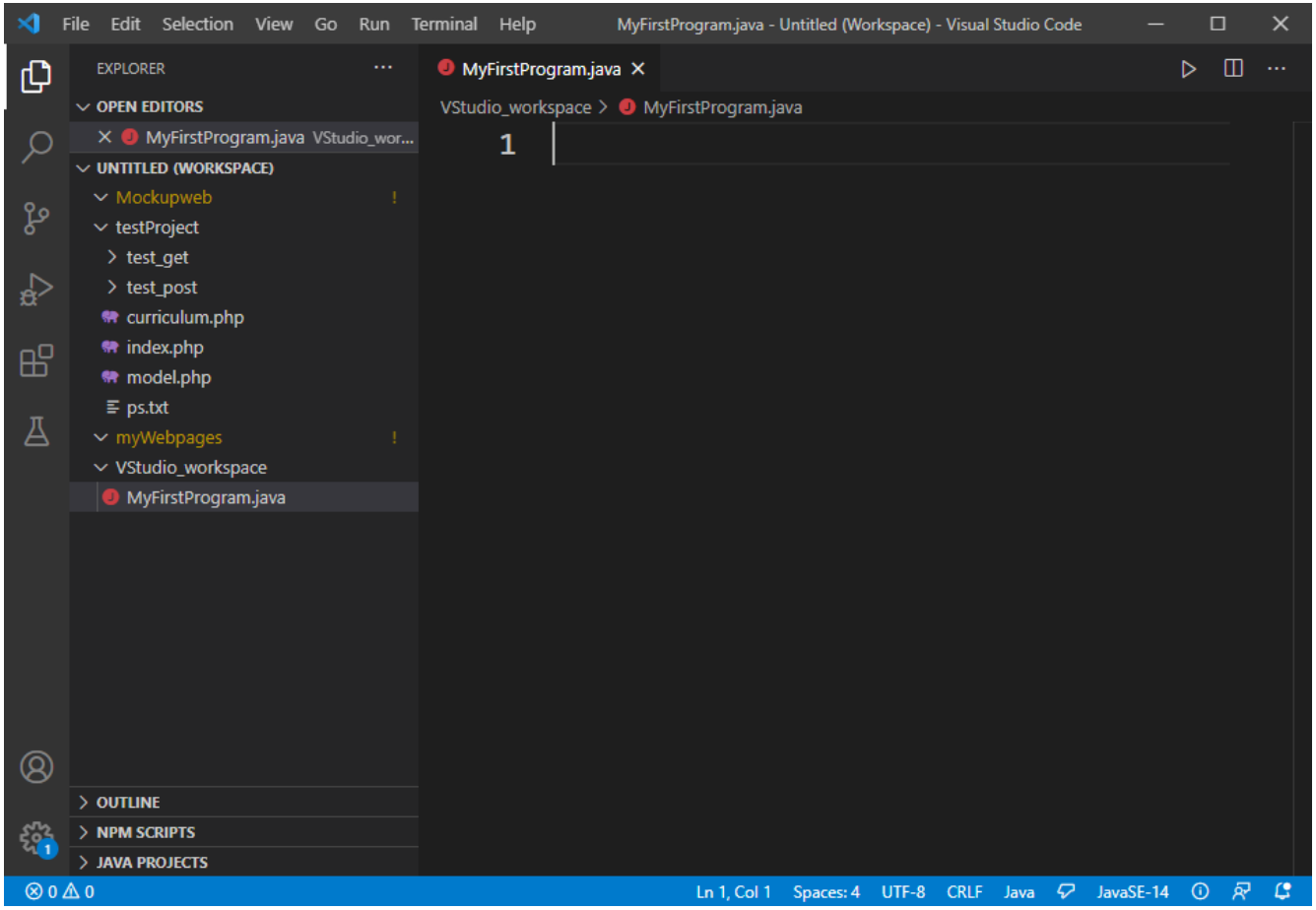
LAB 1

Introduction to Java, Datatypes and Operations

Exercise 0 – Create a Program

- Using Visual Studio Code



Exercise 1: Program Structure

- Write a program structure by
  - Declaring class header
  - Declaring main method

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

Exercise 1: Program Structure

- Class header

```
public class MyFirstProgram

```

- Put the scope of class body

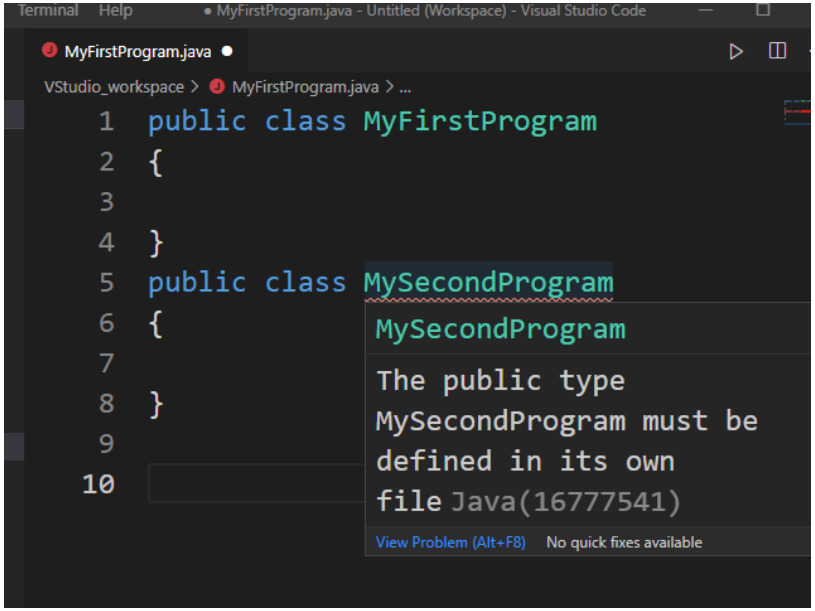
```
public class MyFirstProgram
{

}
```

- You will see that there is no error shown in the editor

Exercise 1: Program Structure

- Declare another class header and put a scope. What do you see in the editor?

```
public class MyFirstProgram
{

}
public class MySecondProgram
{

}
```



Exercise 1: Program Structure

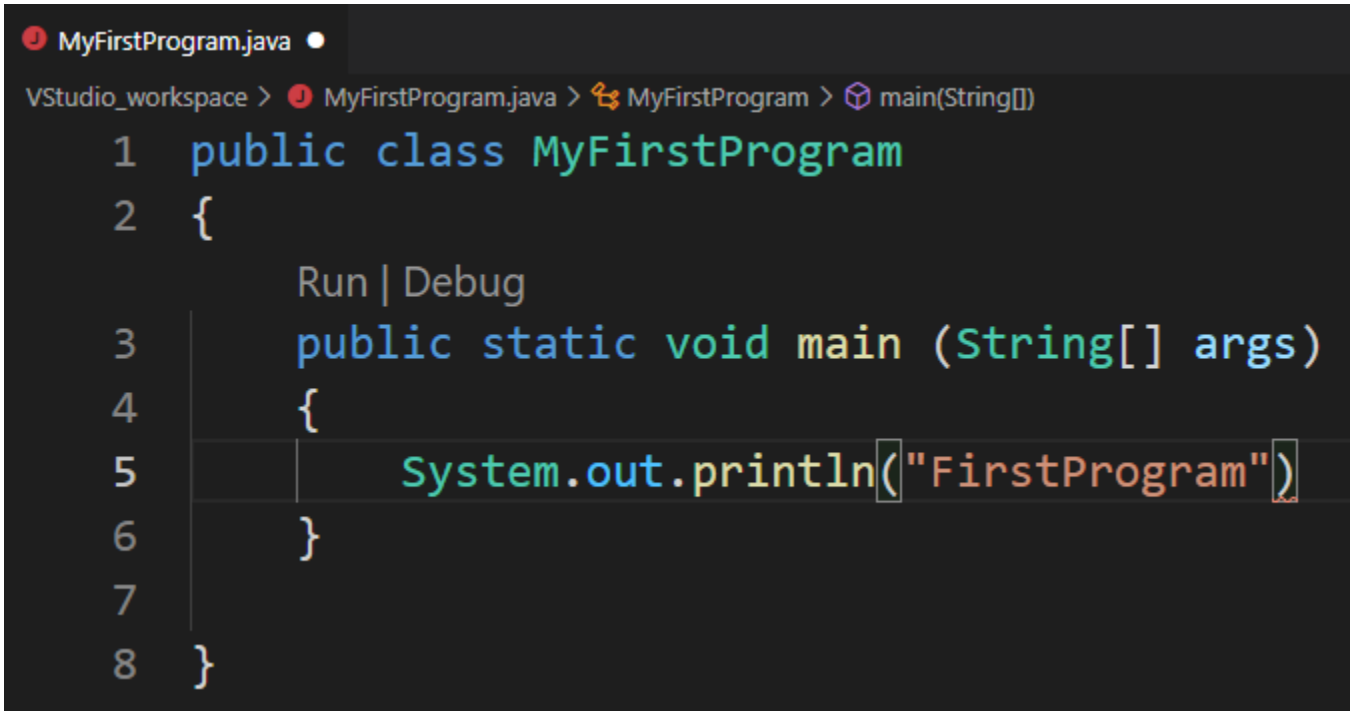
- Delete key word “public” from the second program

```
public class MyFirstProgram
{

}
class MySecondProgram
{

}
```

- What happen?
  - No error
  - Conclusion: you can have more than one class in a java file, but you can make only one class as a public class and save the file with the public class name

Exercise 2: Basic Syntax & Commands

- Remove semicolon (;), then save



- Error : ‘;’ expected
  - Conclusion: a statement need to end with a semicolon (;)
- Put semicolon back, then save and compile

Exercise 2: Basic Syntax & Commands

- Print command

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

- Try
  - Delete quote (“) – what happen?
    - Compile error
  - Conclusion: The print command needs string input which requires quotes (String)

Exercise 2: Basic Syntax & Commands

- Add new print commands

```
public class MyHelloWorld{
	public static void main(String[] args){
		System.out.println("Hello World");
		System.out.print("How are you Mr. A?");
		System.out.print("How are you Mr. B?");
		System.out.println("How are you Mr. C?");
		System.out.print("How" + " " + "are" + " "
+ "you" + " Mr. D?"); }
}
```

- Answer the following questions
  - What is the different between the command *println* and *print* ?
    - The println command prints with newline, the print command prints without newline
  - What dose the operator “+” do ?
    - Concatenate strings

Exercise 2: Basic Syntax & Commands

- Declare variable, assign a value and print the value

```
public class MyHelloWorld{
	public static void main(String[] args){
		int var = 50;
		System.out.println(var);
	}
}
```

- To declare variable
  - [datatype] [variable name] = [value]

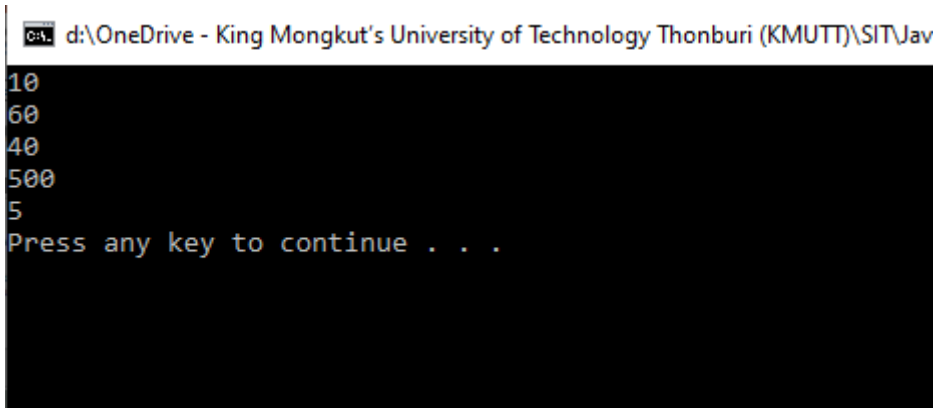
```
int var = 50;
```

- If you put variable as an input for the print command, it will print the value of that variable

Exercise 2: Basic Syntax & Commands

- Print numerical values

```
public class MyHelloWorld{
        public static void main(String[] args){
                System.out.println(10);
                int var = 50;
                System.out.println(var + 10);
                System.out.println(var - 10);
                System.out.println(var * 10);
                System.out.println(var / 10);
        }
}
```



Exercise 3: Comments

- Put some comments in the class MyFirstProgram

```
public class MyFirstProgram
{
        //This is my First Class
       public static void main (String[] args)
       {
              System.out.println("FirstProgram");
              System.out.println("Hello Java");
              System.out.println("I love Java");
       }
}
```

- Put comment (//) in front of the second statements
  - “Hello Java” is not printed

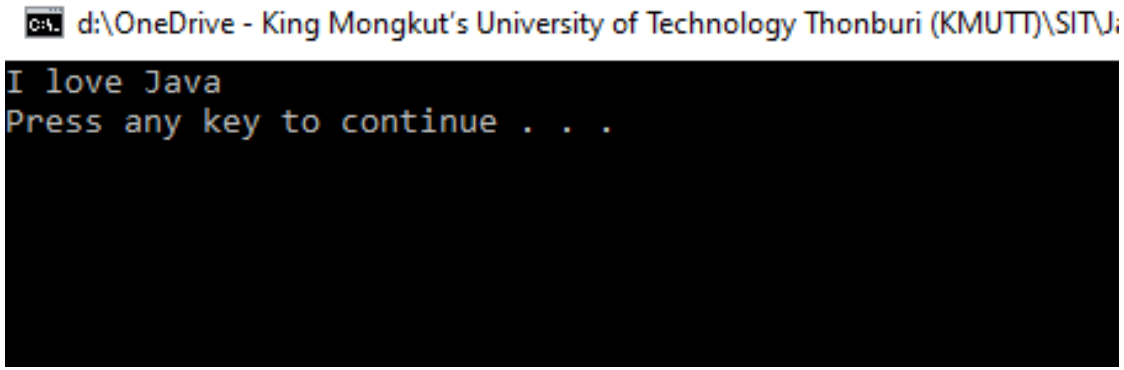
Exercise 3: Comments

- Comment multiple line

```
public class MyFirstProgram
{
        //This is my First Class
        public static void main (String[] args)
        {
                /*System.out.println("FirstProgram");
                System.out.println("Hello Java");*/
                System.out.println("I love Java");
        }
}
```

- What does it print?

Exercise 3: Comments



- Put /* in front of the first line and put */ at the end of the last line where you want to comment

Exercise 4: Identifier and Data types

- Naming
  - Class name – Title case
    - StudentAccount, Student, HelloWorld
  - Constance – Upper case
    - TAX, PI, GRAVITY
  - Variable, method, attribute, and object – camelCase
    - studentName, studentID, weight, height

Exercise 4: Identifier and Datatypes

- Primitive Datatypes

Type	Size	Default	Value Ranges	Contains
boolean	16 bits	false		true or false
byte	8 bits	0	-128 to 127	Signed integer
char	16 bits	\u0000		Unicode character
short	16 bits	0	-32,768 to 32,767	Signed integer
int	32 bits	0	-2,147,483,648 to 2,147,483,647	Signed integer
long	64 bits	0	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807	Signed integer
float	32 bits	0.0	Approximately -3.4E+38 to 3.4E+38 with 7 significant digits	IEEE754 floating point
double	64 bits	0.0	Approximately -1.7E+308 to 1.7E+308 with 15 significant digits	IEEE754 floating point

Exercise 4: Identifier and Datatypes

- Declare a variable b of type byte and assign a value 129, then print b

```
public class MyFirstProgram
{
        public static void main (String[] args)
        {
                byte b = 129;
                System.out.println(b);
        }
}
```

- what happen?

Exercise 4: Identifier and Datatypes

- Declaring datatype out of value range will cause an error

```
public static void main (String[] args)
{
    byte b = 129;
    System.out.println(b);
}
```

- But not always, try this !

```
public class MyFirstProgram
{
        public static void main (String[] args)
        {
                int intMax = 2147483647;
                int operand = 10;
                int result = intMax + operand;
                System.out.println(result);
        }
}
```

Exercise 4: Identifier and Datatypes

- Try and answer

```
long x = 20L + 010L;
System.out.println(x);
```

```
long x = 20L + 0xf;
System.out.println(x);
```

```
long x = 20 + 30;
System.out.println(x);
```

```
int x = 20L + 0xf;
System.out.println(x);
```

```
int x = (int) 20L + 0xf;
System.out.println(x);
```

Casting Conversion

Exercise 4: Identifier and Datatypes

- Character

```
char c = 'A';
char cCode = 65;
System.out.println(c);
System.out.println(cCode);
```

ASCII Code

```
char c = 'A';
char cCode = '\u0041';
System.out.println(c);
System.out.println(cCode);
```

Unicode

Exercise 4: Identifier and Datatypes

- Boolean

```
boolean x = true;
boolean y = false;
System.out.println(x);
System.out.println(y);
```

```
if(x){
     System.out.println("x is True");
}
else{
     System.out.println("x is not True");
}
```

Exercise 5: Escape Sequence

- Print the following escape sequence

Escape Sequence	Name
`\b`	backspace
`\t`	tab
`\n`	newline
`\r`	carriage return
`\"`	double quote
`\'`	single quote
`\\`	backslash

Exercise 6: Operator

- Arithmetic

```
System.out.println(10 + 5);
System.out.println(10 - 5);
System.out.println(10 * 5);
System.out.println(10 / 5);
System.out.println(10 % 5);
```

Exercise 6: Operator

- Relational

```
int x = 10;
int y = 5;
int z = 5;
```

```
if(x>y){
     System.out.println("x is greater than y");
}
else if(x==y){
     System.out.println("x is equal to y");
}
else if(x<y){
     System.out.println("x is less than y");
}
```

Exercise 6: Operator

```
if(x>y){
      System.out.println("x is greater than y");
}
else if(x!=y){
      System.out.println("x is not equal to y");
}
else if(x<y){
      System.out.println("x is less than y");
}
```

- Change the first condition to x!=y, what happen?

Exercise 6: Operator

- Logical

```
int x = 10;
int y = 5;
int z = 5;
```

```
if(x>y || x==y){
      System.out.println("x is greater than or equal to y");
 }
else if(x<y){
      System.out.println("x is less than y");
 }
```

```
if(x>y && x==y){
      System.out.println("x is greater than or equal to y");
 }
else if(x<y){
      System.out.println("x is less than y");
 }
```