Operator Type	Category	Precedence
Unary	postfix	*expr++ expr--*
	prefix	*++expr --expr +expr -expr ~ !*
Arithmetic	multiplicative	* / %
	additive	+ -
Shift	shift	<< >> >>>
Relational	comparison	< > <= >= instanceof
	equality	== !=
Bitwise	bitwise AND	&
	bitwise exclusive OR	^
	bitwise inclusive OR	|
Logical	logical AND	&&
	logical OR	||
Ternary	ternary	? :
Assignment	assignment	= += -= *= /= %= &= ^= |= <<= >>= >>>=

IN THE NAME OF ALLAH

بسم الله الرحمن الرحيم

THE GRACIOUS, THE MERCIFUL.

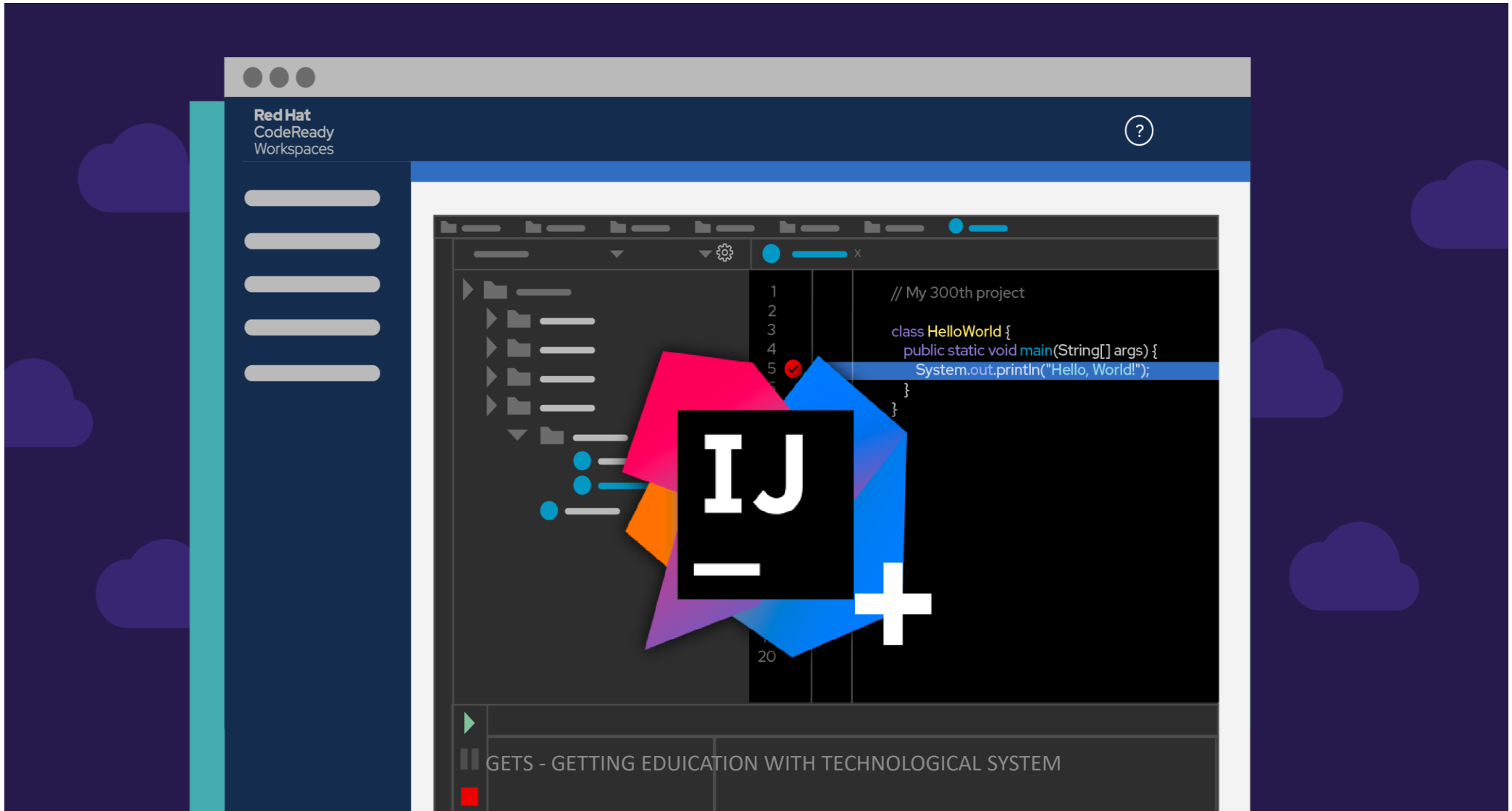
LECTURE: 4
JAVA OPERATORS

GETS - GETTING EDUCATION WITH TECHNOLOGICAL SYSTEM

Instructor: Sir A.Rehman Ali Brohi

Operator Type	Category	Precedence
Unary	postfix	*expr++ expr--*
	prefix	*++expr --expr +expr -expr ~ !*
Arithmetic	multiplicative	* / %
	additive	+ -
Shift	shift	<< >> >>>
Relational	comparison	< > <= >= instanceof
	equality	== !=
Bitwise	bitwise AND	&
	bitwise exclusive OR	^
	bitwise inclusive OR	|
Logical	logical AND	&&
	logical OR	||
Ternary	ternary	? :
Assignment	assignment	= += -= *= /= %= &= ^= |= <<= >>= >>>=

We will use IntelliJ Idea IDE for JAVA Programming



Download https://www.jetbrains.com/idea/download/#section=windows

jetbrains.com/idea/download/#section=windows

IntelliJ IDEA

What's New Features Resources Buy Download

Download IntelliJ IDEA

Windows macOS Linux

Ultimate

For web and enterprise development

Download .exe

Free 30-day trial

Community

For JVM and Android development

Download .exe

Free, built on open source

Version: 2021.2
Build: 212.4746.92
27 July 2021

Release notes ↗

System requirements

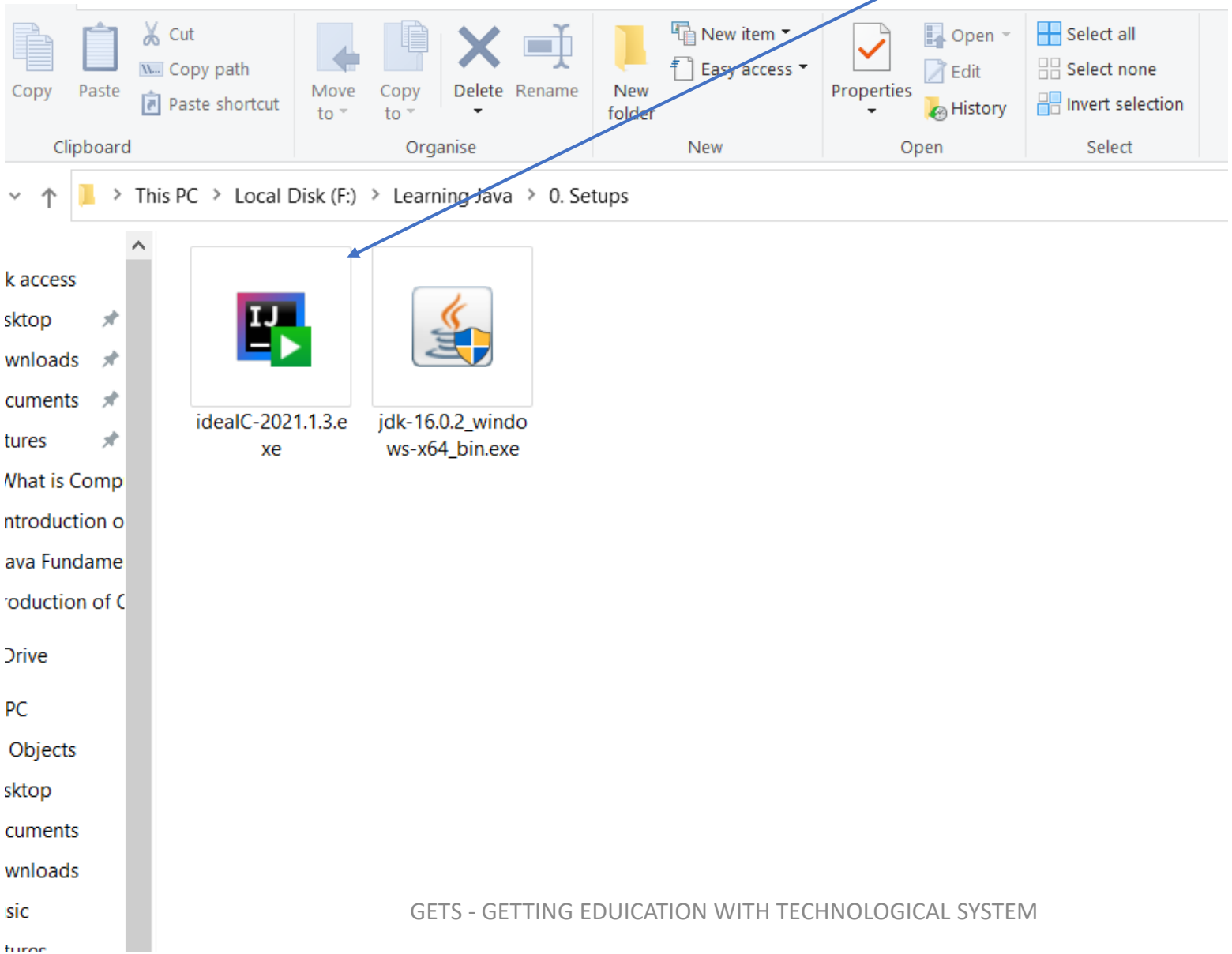
Installation Instructions

Other versions

Third-party software

	IntelliJ IDEA Ultimate	IntelliJ IDEA Community Edition
Java, Kotlin, Groovy, Scala	✓	✓
Android	✓	✓

But we have already have at 0.Setups folder

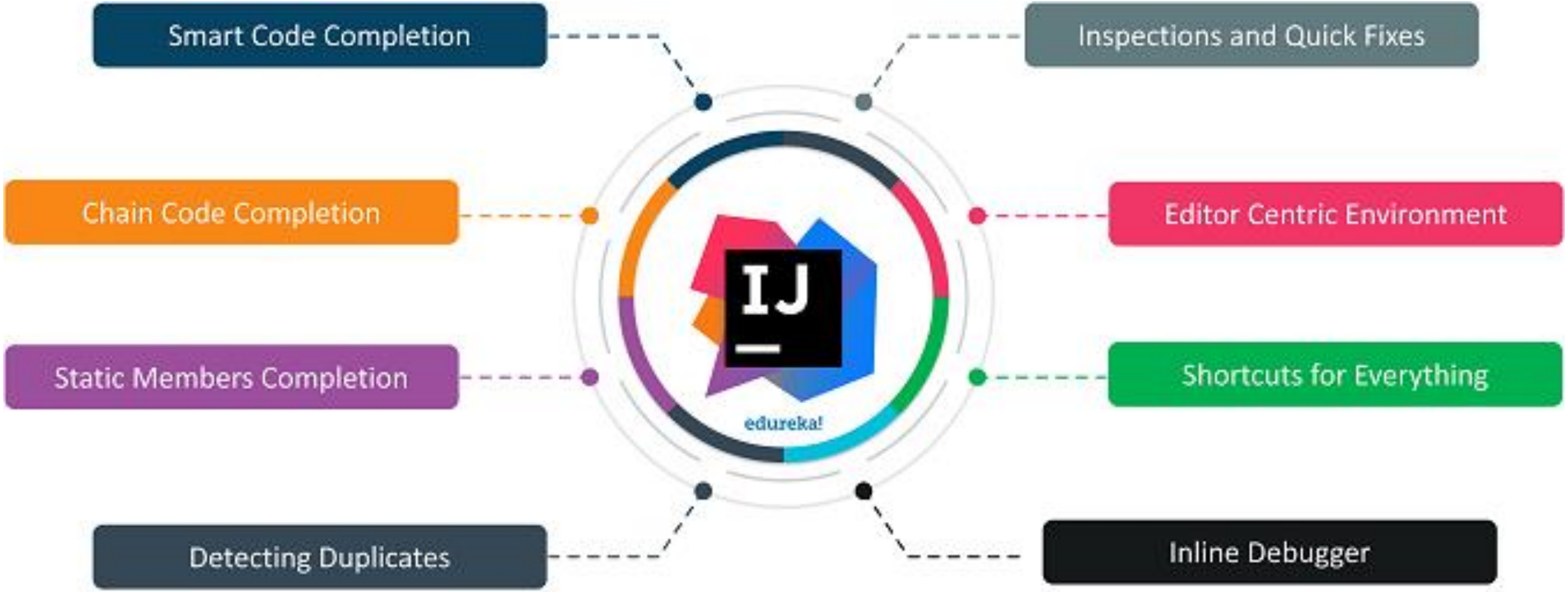


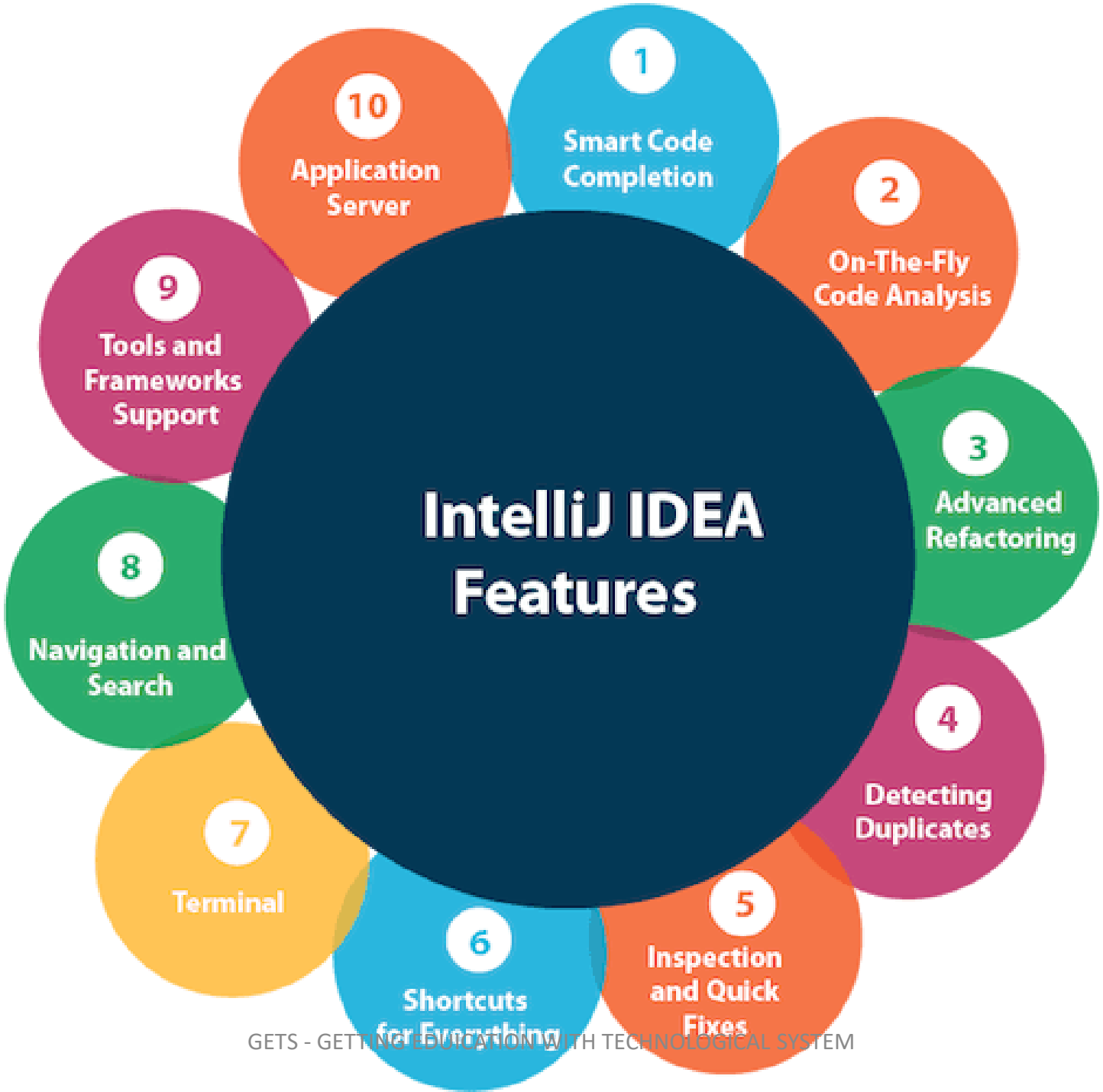
What is IDE?

An Integrated Development Environment (IDE) is a software application that provides comprehensive facilities to computer programmers for software development.

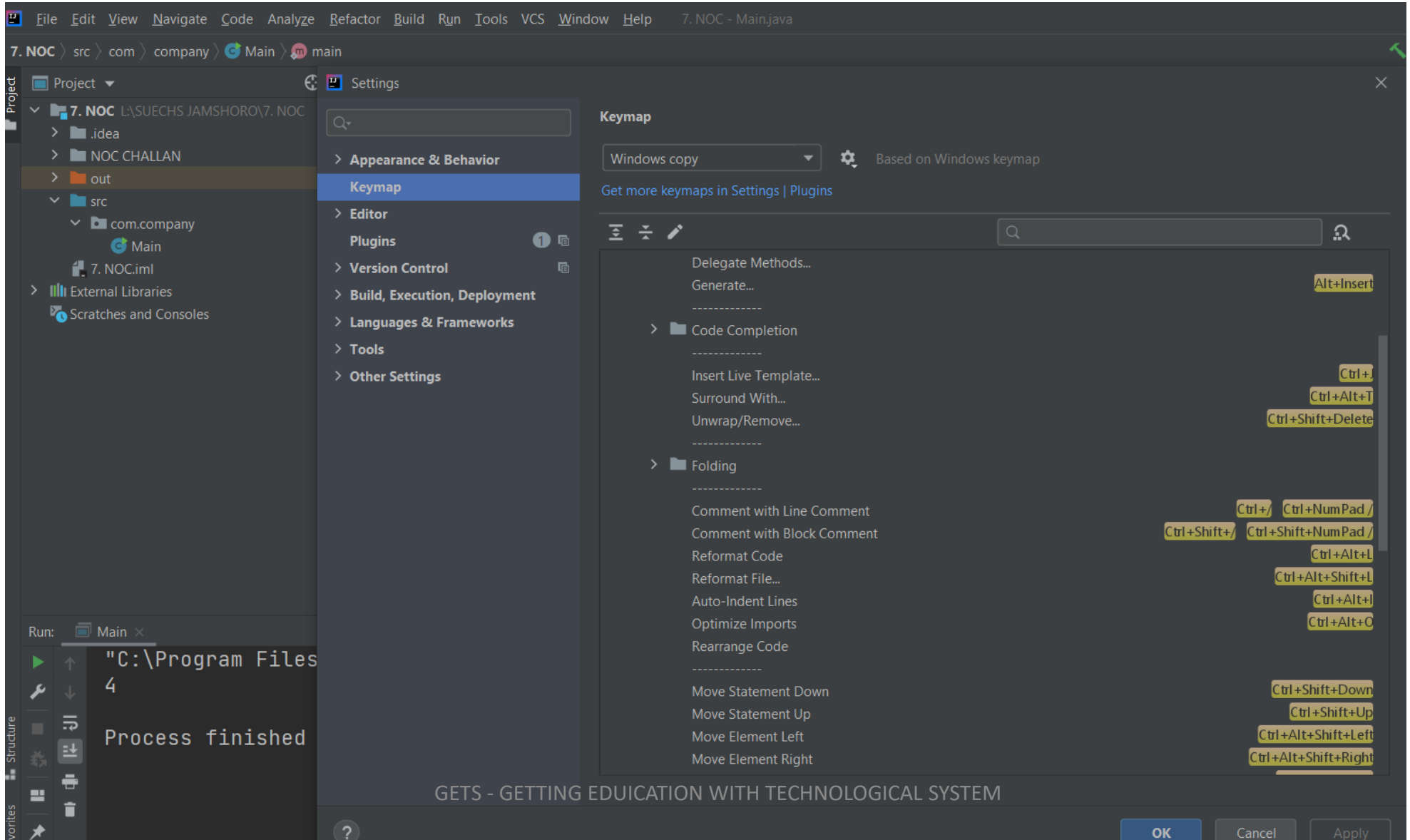
Why IDE?

To reduce the coding time.


Smart Code Completion
Inspections and Quick Fixes
Chain Code Completion
Editor Centric Environment
IJ
edureka!
Static Members Completion
Shortcuts for Everything
Detecting Duplicates
Inline Debugger


IntelliJ IDEA Features
1 Smart Code Completion
2 On-The-Fly Code Analysis
3 Advanced Refactoring
4 Detecting Duplicates
5 Inspection and Quick Fixes
6 Shortcuts for Everything
7 Terminal
8 Navigation and Search
9 Tools and Frameworks Support
10 Application Server

Go to File > setting > keymap > main menu change key setting as you wish



Some shortcuts for Intellij Idea for Code

- Type psvm hit enter for public static void main method
- Sout for System.out.println method

1. Java Unary Operators | ++, --, ~, !

Main.java

```java
package com.company;

public class Main {
    public static void main(String[] args) {
        //unary operators ++, --
        int a = 10;

        System.out.println(a++); //10
        System.out.println(++a); //12
        System.out.println(a--); //12
        System.out.println(--a); //10
    }
}
```

1. Unary Operators | ~ , !

Main.java

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

        //unary operators ~, !
        int a = 10;
        int b = -20;
        boolean c = true;
        boolean d = false;

        System.out.println(~a);//-11 --- minus of total positive value which starts from 0
        System.out.println(~b);//19 --- positive of total minus, positive start from 0
        System.out.println(!c);//false --- opposite of boolean value
        System.out.println(!d);//true --- opposite of boolean value
    }
}
```

2. Java Arithmetic's | *, /, %, -, +

Main.java

```java
package com.company;

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

        //Java Arithmetics *, /, %, -, +
        int a = 10;
        int b = 5;

        System.out.println(a+b); //15
        System.out.println(a-b); //5
        System.out.println(a/b); //2
        System.out.println(a%b); //0
        System.out.println(a*b); //50
    }
}
```

Main.java

```java
package com.company;

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

        //Java Arithmetics *, /, %, -, +

        System.out.println(10*10/5+3-1*4/2); //21
    }
}
```

3. Java Left Shift Operator | <<

Main.java

```java
package com.company;

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

        //Java Left shift operators <<

        System.out.println(10<<2); //10*2^2=10*4=40
        System.out.println(10<<3); //10*2^3=10*8=80
        System.out.println(20<<2); //20*2^2=20*4=80
        System.out.println(15<<4); //15*2^4=15*16=240
    }
}
```

3. Java **Right Shift** | **>>**

Main.java

```java
package com.company;

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

        // right shift operator

        System.out.println(10>>2); //10/2^2=10/4=2
        System.out.println(20>>2); //20/2^2=20/4=5
        System.out.println(20>>3); //20/2^3=20/8=2

    }
}
```

4. Java AND Operator logical **&&** and Bitwise **&**

Main.java

```java

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

        //Java AND Operator logical && and Bitwise &
        //logical && if 1st condition is false it will not check the 2nd condition
        //Bitwise & if 1st condition is false it will also check the 2nd condition

        int a = 10;
        int b = 5;
        int c = 20;

        System.out.println(a<b&&a<c); //false && true = false
        System.out.println(a<b&a<c); //false & true = false

    }
}
```

4. Java AND Operator logical **&&** and Bitwise **&** **with increment method**

Main.java

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

        //Java AND Operator logical && and Bitwise &
        //logical && if 1st condition is false it will not check the 2nd condition
        //Bitwise & if 1st condition is false it will also check the 2nd condition

        int a = 10;
        int b = 5;
        int c = 20;

        System.out.println(a<b&&a++<c); //false && true = false
        System.out.println(a); //10 because 2nd condition is not checked
        System.out.println(a<b&a++<c); //false & true = false
        System.out.println(a); //11 because 2nd condition is also checked

    }
}
```

4. Java OR Operator logical || and Bitwise |

Main.java

```java
package com.company;

public class Main {
    public static void main(String[] args) {
        int a = 10;
        int b = 5;
        int c= 15;

        System.out.println(a>b||a>c); //true || false = true
        System.out.println(a<b|a<c); //false | true = true


    }
}
```

4. Java OR Operator logical || and Bitwise | with increment method

Main.java

```java
package com.company;

public class Main {
    public static void main(String[] args) {
        int a = 10;
        int b = 5;
        int c= 15;

        System.out.println(a>b||a++>c); //true || false = true
        System.out.println(a); //10 because 2nd condition not checked
        System.out.println(a<b|a++<c); //false | true = true
        System.out.println(a); //11 because 2nd condition is checked


    }
}
```

4. Java Bitwise exclusive **XOR** ^

Main.java

```java
package com.company;

public class Main {
    public static void main(String[] args) {
        int a = 10;
        int b = 5;
        int c= 15;

        System.out.println(a<b^a>c); //false ^ false = false
        System.out.println(a>b^a>c); //true  ^ false = true
        System.out.println(a<b^a<c); //false ^ True  = true
        System.out.println(a>b^a<c); //true  ^ True  = false



    }
}
```

Logical AND Operator (& and &&)

Operand 1	Operand 2	Returned Value
False	False	False
False	True	False
True	False	False
True	True	True

Logical OR Operator (| and ||)

Operand 1	Operand 2	Returned Value
False	False	False
False	True	True
True	False	True
True	True	True

Logical XOR Operator (^)

Operand 1	Operand 2	Returned Value
False	False	False
False	True	True
True	False	True
True	True	False

4. Java Ternary operators | ? , :

7. NOC > src > com > company > Main > main

Main.java

```java
package com.company;

public class Main {
    public static void main(String[] args) {
        //Java Ternary Operator use

        int a = 10;
        int b = 5;
        int minimum = (a<b)?a:b;


        System.out.println(minimum);//if 1st condition is true then it print the minimum num
    }
```

Run: Main

```
"C:\Program Files\Java\jdk-16.0.2\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Communi
5

Process finished with exit code 0
```

4. Java Assignment operators | =, +=, -=, *=, /=, %=, &=, ^=, |=, <<=, >>=, >>>=

7. NOC > src > com > company > Main > main

Main.java

```
4   public static void main(String[] args) {
5       //Java assaignment operator
6
7       int a = 10;
8       int b = 5;
9       a+=3; //a=a+3 (a=10+3)
10      b-=2; //b=b-2 (b=5-2)
11
12      System.out.println(a);
13      System.out.println(b);
14  }
15 }
```

Run: Main

```
"C:\Program Files\Java\jdk-16.0.2\bin\java.exe" "-javaagent:C:
13
3

Process finished with exit code 0
```

7. Java Relational operators | <, >, <=, >=, instanceof

```
public static void main(String[] args) {
    //Java Relational operator

    int number1 = 10;
    int number2 = 5;


    System.out.println(number1>number2);

}
}
```

```
"C:\Program Files\Java\jdk-16.0.2\bin\java.exe" "-javaagent:C:\Program Files\JetBra
true

Process finished with exit code 0
```

7. NOC > src > com > company > Main > main

Main.java

```java
public static void main(String[] args) {
    //Java Relational operator

    int number1 = 10;
    int number2 = 5;


    System.out.println(number1>number2);
    System.out.println(number1<number2);
    System.out.println(number1>=number2);
    System.out.println(number1<=number2);

}
```

Run: Main

```
"C:\Program Files\Java\jdk-16.0.2\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\Intelli
true
false
true
false

Process finished with exit code 0
```

7. Java Relational operators | String compression ==

7. NOC > src > com > company > Main > main

Main.java

```
4    public static void main(String[] args) {
5        //Java Relational operator
6
7        String name1 = "Student";
8        String name2 = "Studentt";
9
10
11       System.out.println(name1==name2);
12
13
14   }
15 }
16
```

Run: Main

```
"C:\Program Files\Java\jdk-16.0.2\bin\java.exe" "-javaagent:C:\Program Files\J
false

Process finished with exit code 0
```