

March-2<sup>nd</sup>

→ Kotlin programming

1) For this course we are using IntelliJ IDE for learning course.

i) Install community version of IntelliJ IDE

ii) Open IntelliJ IDE & Set up the JDK (Java-Development Kit)

iii) SDK (Software development Kit) By clicking on New project icon, → It will open a new window to download Jdk for the first time and it will be downloaded to create a new project.

\*): Create New project by selecting the programming language as 'Kotlin & IntelliJ IDE'

\*) Q) How to check IntelliJ updates?

A) Help → Tools → Check Updates

B) Go to tools → Check Updates, It will pop up a new window for IntelliJ updates. If there is an update then install it otherwise No update.

\*) Q) How to check Kotlin updates?

A) Go to tools → Kotlin → Configure Kotlin updates

\*) Create a new kotlin file under src folder & Name it accordingly  
accordingly  
as given name

code in the newly created file:

Initially:

```
fun main() {
```

} → main function which is the entry point of our program

fun : It is keyword for creating a function

main : main is the name of the function & main is the special function because it is the entry point of the function.

→ We can give a different name to the functions, But if we try to change the name of the main function which by default "main", then we cannot see run option to run our program.

(1) (parenthesis) → It's important because it is part of syntax & tells that it is function.

\*)- fun main()

Println("Hello")  
↳

① Printline (Println) is a function that takes input & output that in the console.  
② It is built-in function

↓  
Input  
we have passed into  
Parenthesis.

Kotlin Source  
code (Human-readable  
code)

?DKS  
[Kotlin Compile]

Java bytecode code.  
(at this stage, it is  
not converted into  
machine code.)

↓  
PVM-HMC  
↳ Windows

↓  
JVM-HMC  
↳ Macos

This is how Java & Kotlin can achieve "Write once and run everywhere" this is because the work of converting that MC

MC (Machine code) which a specific platform can understand has shifted from programmer to the platform which implements JVM

\* you can add Java code to the existing Kotlin program & vice versa.  
They are inter-operable.

### Variables:-

Variables are like data containers which are used to hold data values.

### To create a variable:

- (i) `Var UserName: String = "puneeth" // Var UserName = "Alex";` [The type of data.]
- (ii) `Val name: String = "Pohilla" // Val age = 22;`
- (iii) Kotlin is statically typed language. That means, it will check at type of variables at compile time not at run time.
- (iv) If I declare a variable as 'val' we cannot change the value of the variable and vice versa with 'var'
- Q) What about type casting in Kotlin?
- String → It is basically a data type that stores the sequence of characters.
- Int → It is a data type that is used to store negative & positive numbers but has limit (i.e.) range on how much value it can store.

Q) Difference b/w float & double?

A) They are different in precision. In which double has more precision than float.

→ \*) Code :-

`val g = 2F;` → explicitly mentioned it is a float but we have not assigned

`val h = 2;` fractional part

`println("g,h")`; → O/P: 2.0, 2.

\*) val h : Double = 2 "error (Type mismatch)"

It's because Kotlin requires explicit type conversion

→ val h : Double = 2.0 → correct

\*) char & Boolean data type :-

1). Boolean represents only one bit of information either true or false.  
The boolean type in Kotlin is the same as java. These operations disjunction (||) or conjunction (&&) can be performed on boolean types.

character data type :-

\*) character data type represents the small letters 'a' to 'z', capital letters ('A' to 'Z'), digits (0-9) & other symbols.

In Kotlin the 'char' type is designed to represent a single 16-bit Unicode character which means that each 'char' type variable can represent only one Unicode character. It can be a letter

a digit, a symbol, or any other unicode character with in range 0 to 65,535

→ Unicode is standard that assigns a unique number value to every character

→ Imagine a text which is sequence of characters and each character is assigned a unique number, these numbers are stored using 16 bits.

Esc:                      0x0041    0x006c  
String(characters): H    E    L    L    O  
Unicode    0x0041 0x0065 0x006c

→ In Kotlin all data types are objects.  
short, byte, Int, long → contains Integer values.

\* :-