1. Why Kotlin is preferred over other languages for Android application Development?
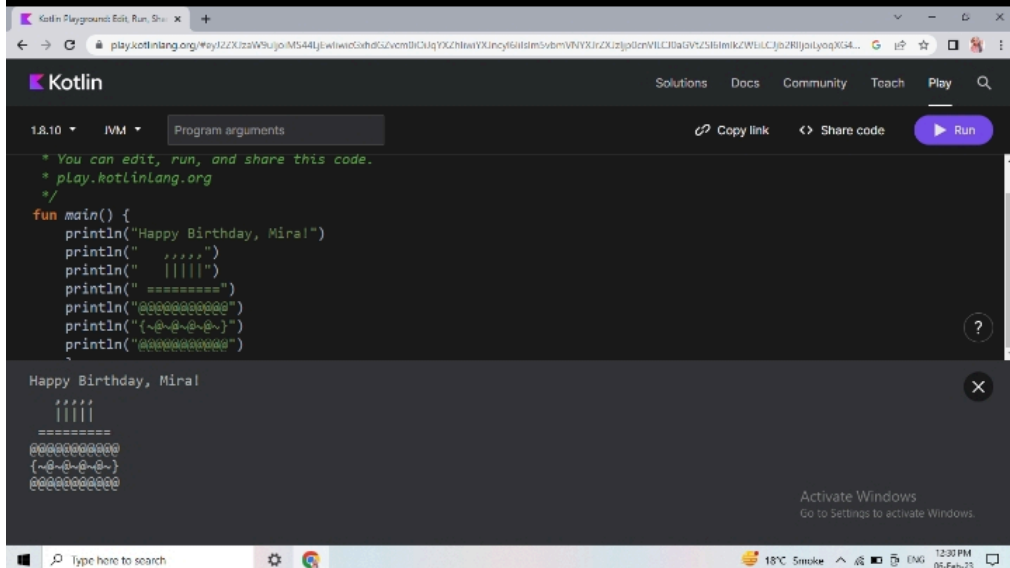
Solution →

Kotlin is a programming language, that has been available as a stable release since 2016. Because of its rapidly increasing popularity over the past 5 years — it is already the number 1 choice for Android Development Because of its following features →

1. Shorter program for the same Task.
2. Easy code.
3. Java compatibility
4. Eliminating Null References.
5. provide solution To some of Java flaws
6. Open Source Technology.
7. compiles to Jvm bytecode.

Q2 Kotlin program →.

12:30

Kotlin Playground: Edit, Run, Sha

play.kotlinlang.org/#eyJ2ZXJzaW9uIjoiMS44LjEwIiwicGxhdGZvcm0iOiJqYXZhIiwiYXJncyI6IiIsIm5vbmVVNYXJnZXJjdjpp0cnVlLCJ0aGVtZSI6ImlkZWEiLCJjb2RlIjoiLyoqXG4...

Kotlin   Solutions   Docs   Community   Teach   **Play**   🔍

1.8.10 ▾   JVM ▾   Program arguments   🔗 Copy link   ‹› Share code   ▶ Run

```
 * You can edit, run, and share this code.
 * play.kotlinlang.org
 */
fun main() {
    println("Happy Birthday, Mira!")
    println("   ,,,,,")
    println("   |||||")
    println(" =========")
    println("@@@@@@@@@@")
    println("{~@~@~@~@~}")
    println("@@@@@@@@@@")
```

Happy Birthday, Mira!   ✕
   ,,,,,
   |||||
 =========
@@@@@@@@@@
{~@~@~@~@~}
@@@@@@@@@@

Activate Windows
Go to Settings to activate Windows.

Type here to search

2. How to Declare a variable in kotlin? Difference b/w Val and var by giving Suitable examples.

Solution →

Kotlin uses two different keywords to declares variables : <u>val</u> and <u>var</u>

- <u>Syntax to declare variable</u> :

var name : String = "Anshu"

↑ Variable Name   ↑ Type of Variable   ↑ Value of Variable

→ <u>variable</u> are container used to store data

VAL (Value)	VAR (value)
→ VAL is immutable & final.	→ VAR is called mutable & Non-final variables.
→ The object stored using val keyword cannot be changed, it cannot be reassigned throughout the program.	→ The value of variable that is declared using var can be changed anytime throughout the program.

Example — VAL (value)

```
fun main ()
{
    val marks = 10
    println(" previous marks is ${marks} "):

    marks = 30
    println(" updated marks is ${marks}")
}
```

> Val cannot be reassigned

Example — VAR (value)

```
fun main ()
{
    var marks = 30
    println(" previous marks is ${marks}")
    marks = 40
    println(" Updated marks is ${marks}")
}
```

> previous marks is 30
> updated marks is 40