# Day-1, Assignment-2

Task 1: Set up the Kotlin development environment and write a simple Kotlin script to validate the setup

fun main() {

val name = "Kotlin"

println("Hello, " + name + "!")

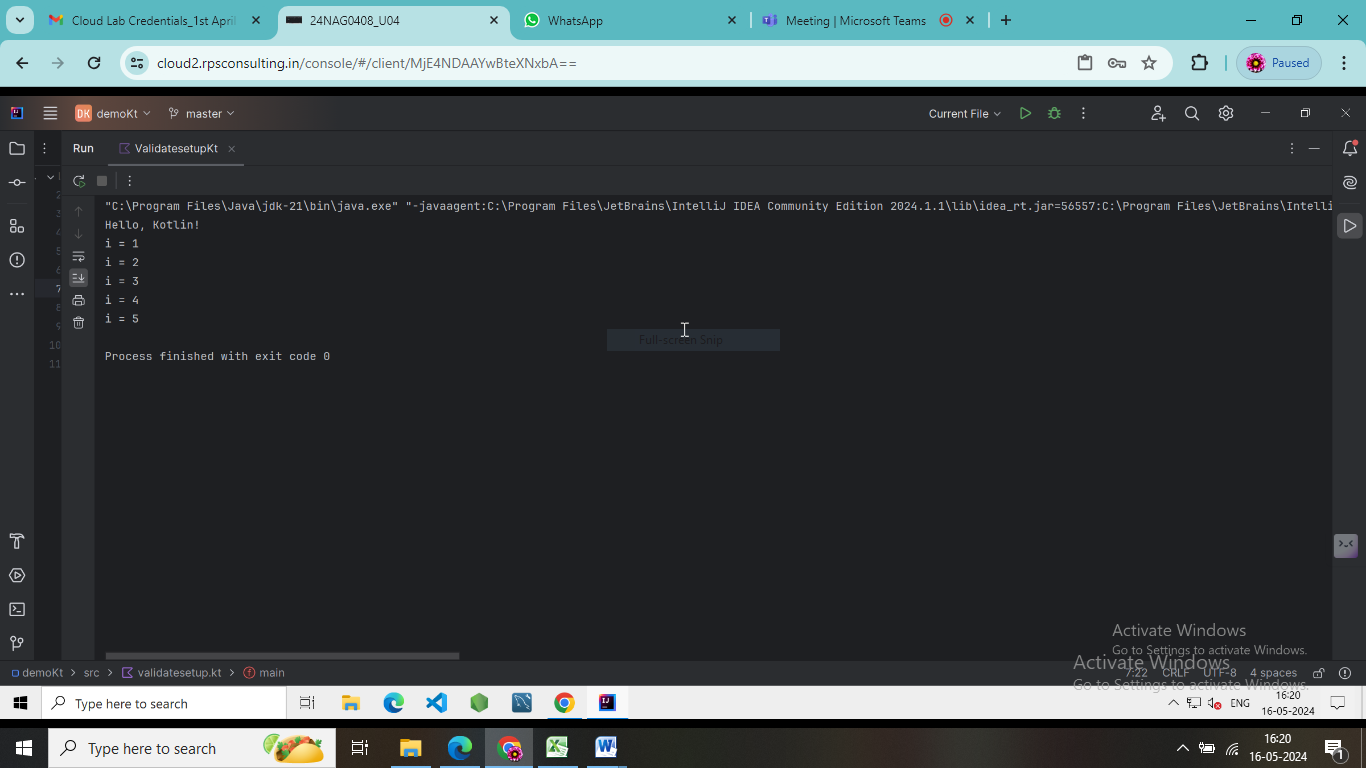
for (i in 1..5) {

println("i = $i")

}

}

## Output:



Task 2: Experiment with Kotlin's string templates to create dynamic welcome messages.

fun main() {

val userName = "Jaanu"

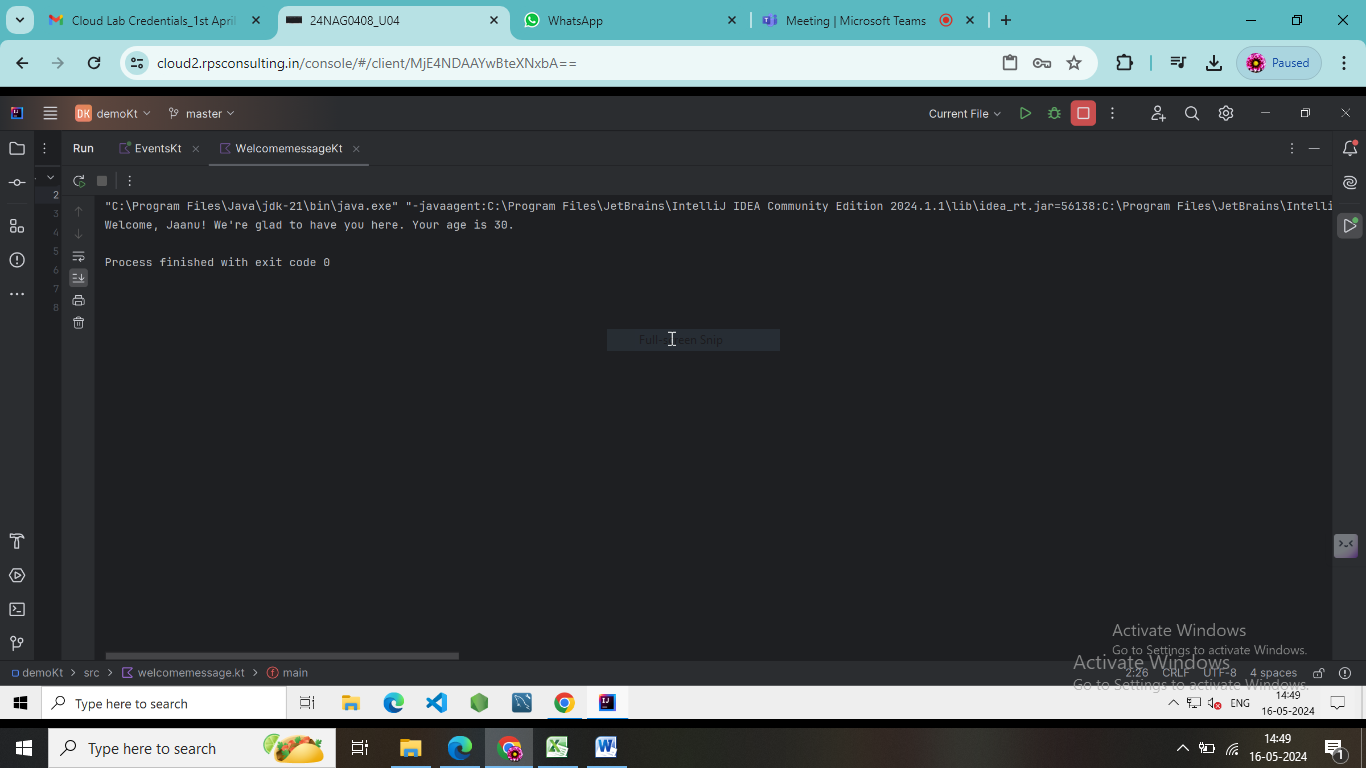
val userAge = 30

val welcomeMessage = "Welcome, $userName! We're glad to have you here. Your age is $userAge."

println(welcomeMessage)

}

## Output:



Task-3: Define data types to represent event details such as name, date, and attendee count.

import java.time.LocalDate

// Define a data class to represent event details

data class Event(

val name: String,

val date: LocalDate,

val attendeeCount: Int

)

fun main() {

// Creating an instance of Event

val event = Event("Birthday Party", LocalDate.of(2024, 5, 20), 50)

// Accessing properties of the event

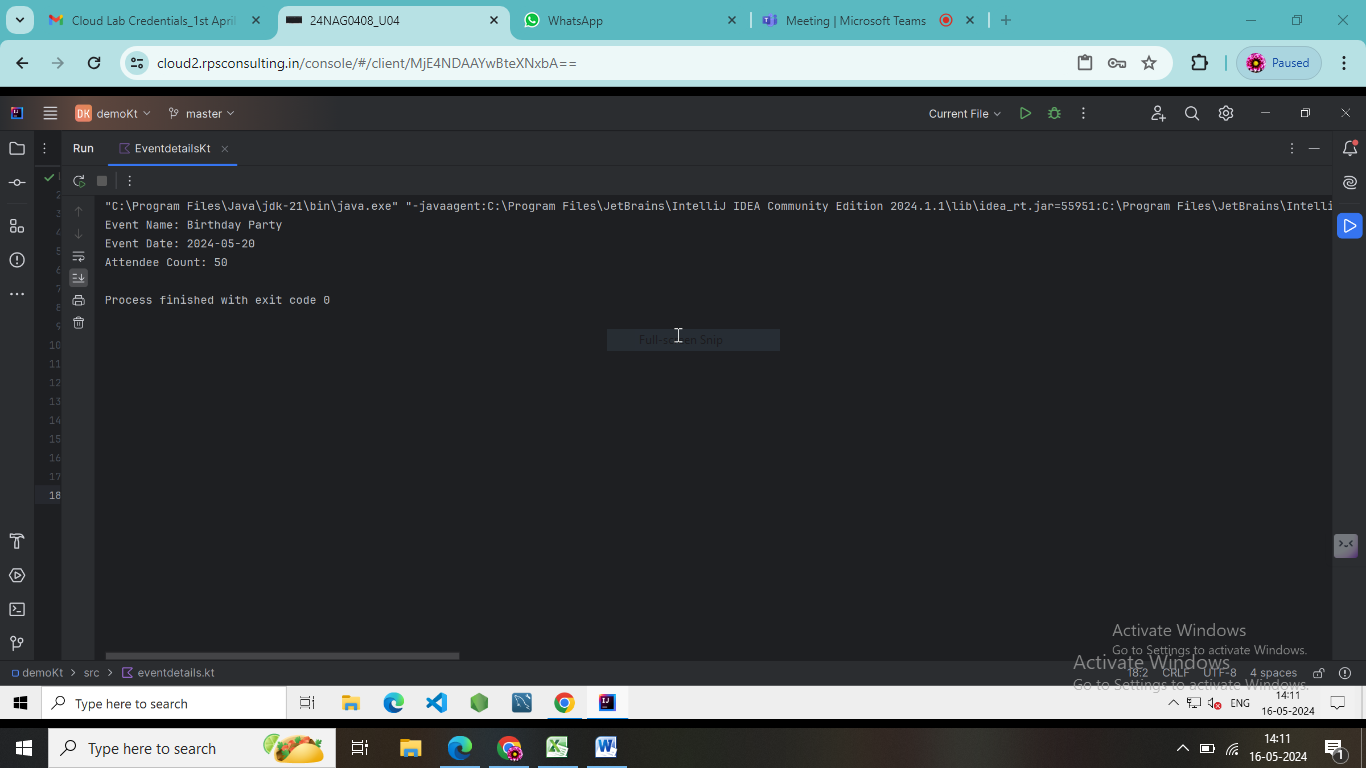
println("Event Name: ${event.name}")

println("Event Date: ${event.date}")

println("Attendee Count: ${event.attendeeCount}")

}

## Output:



Task-4: Implement a basic user input flow to create new events using if and when statements.

fun main() {

println("Welcome to the Event Creator!")

val events = mutableListOf<String>()

while (true) {

println("\nMenu:")

println("1. Create a new event")

println("2. View all events")

println("3. Exit")

print("Please enter your choice: ")

val choice = readLine()?.toIntOrNull()

when (choice) {

1 -> {

print("Enter the name of the event: ")

val eventName = readLine()

events.add(eventName ?: "Unnamed Event")

println("Event '$eventName' created successfully!")

}

2 -> {

if (events.isEmpty()) {

println("No events created yet.")

} else {

println("All Events:")

events.forEachIndexed { index, event ->

println("${index + 1}. $event")

}

}

}

3 -> {

println("Exiting the Event Creator. Goodbye!")

return

}

else -> {

println("Invalid choice. Please enter a number between 1 and 3.")

}

}

}

}

## Output:

