1.write a kotlin program that takes a nullable integer as input and print its square if it is not null, or "input is null " otherwise

fun main() {

val input: Int? = readLine()?.toIntOrNull()

if (input != null) {

println("Square of the input is: ${input \* input}")

} else {

println("Input is null")

}

}

2. implement a number guessing game in which the user is promoted to enter a number between 1 and 100 until he or she guesses correctly After every wrong guess,the user is told whether the guess was too high or too low

import kotlin.random.Random

fun main() {

val targetNumber = Random.nextInt(1, 101)

var guess: Int? = null

println("Welcome to the Number Guessing Game!")

println("Try to guess the number (between 1 and 100):")

while (guess != targetNumber) {

guess = readLine()?.toIntOrNull()

if (guess != null) {

when {

guess < targetNumber -> println("Too low! Try again.")

guess > targetNumber -> println("Too high! Try again.")

else -> println("Congratulations! You guessed the correct number: $targetNumber")

}

} else {

println("Please enter a valid number between 1 and 100.")

}

}

}

3. create a function in kotlin that takes name as input and prints the greeting message make the message customizable and provide a default message if no custom message is provided

fun greet(name: String, greeting: String = "Hello") {

println("$greeting, $name!")

}

fun main() {

greet("Alice")

greet("Bob", "Welcome")

greet("Charlie", "Good day")

}

4. create a kotlin application for rolling die using classes

import kotlin.random.Random

class Die(private val sides: Int = 6) {

fun roll(): Int {

return Random.nextInt(1, sides + 1)

}

}

fun main() {

val die = Die()

println("Rolling the die 5 times:")

repeat(5) {

val result = die.roll()

println("Roll ${it + 1}: You rolled a $result!")

}

}

5. create a kotlin application to demostrate the companion objects,getter and setter properties

class Person(name: String) {

companion object {

private var personCount = 0

fun getPersonCount(): Int {

return personCount

}

init {

personCount++

}

}

private var \_name: String = name

var name: String

get() = \_name

set(value) {

if (value.isNotEmpty()) {

\_name = value

} else {

println("Name cannot be empty")

}

}

}

fun main() {

val person1 = Person("Alice")

val person2 = Person("Bob")

println("Person 1 Name: ${person1.name}")

person1.name = "Alice Smith"

println("Updated Person 1 Name: ${person1.name}")

println("Person 2 Name: ${person2.name}")

person2.name = ""

println("Attempted to update Person 2 Name: ${person2.name}")

println("Total persons created: ${Person.getPersonCount()}")

}