Batch#12/Android Class

Remote Learning Assignment-Week1

1. Try the statement below in REPL. Please answer that there are correct or not and explain the reasons.
2. val id = 9527

ANS: Correct

1. var number: int = 5.6

ANS: Incorrect

WHY: The floating-point literal does not conform to the expected type Int

1. var title = “Hello Kotlin!”

ANS: Correct

1. val content: String = null

ANS: Incorrect

WHY: Null cannot be a value of a non-null type String

1. val number: Double? =556.6

ANS: Correct

1. var list: List<String>? = listOf(null, null)

ANS: Incorrect

WHY: Type inference failed. Expected type mismatch: inferred type is List<Nothing?> but List<String>? was expected

1. var list: List<String?>? = listOf(“Wayne”, “Chen”)

ANS: Correct

1. list. size

ANS: Incorrect

WHY: only safe (?.) or non-null asserted (!!.) calls are allowed on a nullable receiver of type List<String>?

1. list?.size

ANS: Correct

1. ANS: c. “Your grace.”

WHY: Length of b is 39

1. ANS: d. val array = Array(6){1000.0.pow(it)}

**val array = Array(6){1000.0.pow(it)}**

**val units = arrayOf("byte", "KB", "MB", "GB", "TB", "PB")**

**for ((i, value) in array.withIndex()) {**

**println("1 ${units[i]} = ${value.toLong()} bytes\n")**

**}**

4.5.

open class Human(val name: String) {  
 open fun attack() {  
 *println*("<the name passed in> use Fist Attack!")  
 }  
}  
class Mage: Human() {  
 override fun attack(){  
 *println*("<the name passed in> use Fireball!")  
 }  
}  
  
fun main(args: Array<String>) {  
 val humanA = Human("David")  
 humanA.attack()  
 val humanB=Human("Mage")  
 humanB.attack()  
}

6.

open class Human(val name: String) {  
  
 open fun attack() {  
 *println*("<the name passed in> use Fist Attack!")  
 }  
 open fun humanHasMana(){  
 *println*("Human has Mana is true.")  
 }  
}  
class Mage: Human() {  
  
 override fun attack(){  
 *println*("<the name passed in> use Fireball!")  
 }  
 override fun humanHasMana(){  
 *println*("Human has mana is not true.")  
 }  
  
}  
  
fun main(args: Array<String>) {  
 val humanA = Human("David")  
 humanA.attack()  
 humanA.humanHasMana()  
 val humanB = Human("Mage")  
 humanB.attack()  
 humanB.humanHasMana()  
}