Basics of Scala

Vicente De Leon

UID: 2001014594

Assignment 2

*Note- Canvas: don’t forget to mention ONLY the question numbers (not the text) and write the resources you use to solve the problems.*

1. Question 1

Resources (Please check Scala Worksheet Resources for more detail):

* <https://docs.scala-lang.org/overviews/scala-book/two-types-variables.html#:~:text=The%20difference%20between%20val%20and%20var%20is%20that%20val%20makes,as%20values%20rather%20than%20variables>.
* <https://stackoverflow.com/questions/10040886/what-does-it-mean-assign-to-a-field-in-scala>
* <https://www.includehelp.com/scala/this-keyword-in-scala.aspx>
* <https://www.geeksforgeeks.org/scala-string-touppercase-method-with-example/>
* <https://stackoverflow.com/questions/9727637/new-keyword-in-scala>

Observations:

* Setting x to private (mutable) means that it can only be modified or access within class Item. I t helps prevent outsides from accessing the fields directly.
* When using “\_”, it will assign a default value as null, 0, or False.
* Keyword “this” is used to fetch the variable of the class.

A screenshot of a computer program

Description automatically generated with medium confidence

1. Question 2

Resources (Please check Scala Worksheet Resources for more detail):

* <https://leportella.com/scala-iii/>
* <https://www.w3resource.com/python-exercises/class-exercises/python-class-basic-1-exercise-9.php>
* <https://docs.scala-lang.org/overviews/core/string-interpolation.html>

Observations:

* We are using var, just like in question 1, because we want to modify the attributes.
* We are also doing string interpolation using the “$” sign.

A screen shot of a computer

Description automatically generated with medium confidence

1. Question 3

Resources (Please check Scala Worksheet Resources for more detail):

* <https://www.educative.io/answers/how-to-create-a-class-in-scala>
* <https://www.tutorialspoint.com/scala/scala_functions.htm>
* <https://www.geeksforgeeks.org/class-and-object-in-scala/>

Observations:

* Object Main runs in Scala Class (using IntelliJ) not in Scala Worksheet. I had to change name from Object Main to Object Question3 to run the object and get the answers. This might be related to the duplicates in both the worksheet and class.

A screenshot of a computer program

Description automatically generated with medium confidenceA screenshot of a computer program

Description automatically generated with medium confidence

A screen shot of a computer

Description automatically generated with medium confidence

1. Question 4

Resources:

* Base code from: <https://www.geeksforgeeks.org/access-modifiers-in-scala/>

Observations:

* The following code were taking from the above resource in order to implement what was asked.
* Class Private -> has private var x.
* Class Protected -> protected var y. This is similar to private; it can only be accessible from sub classes of the base class in which the member has been defined. Var y can be accessed and modified in “New1” class, which extends Protected.
* Class Public -> can be easily access anywhere.
* We use “e2” to access Pubic var z without any constraint/problem.

A screen shot of a computer

Description automatically generated with medium confidence

1. Question 5

Case classes:

* defined in a single statement with parameters.
* Support pattern matching.
* Can’t be extended by other classes.
* Doesn’t required “new” keyword to create objects of case classes.
* Objects of case class compare the structure of objects.
* Have predefined hashcode and equals methods.

Regular classes:

* Defined by defining methods and fields.
* Doesn’t support pattern matching.
* Can be extended by other classes.
* Keyword “new” is required for creating objects of regular classes.
* Comparison of objects of regular class is done using reference comparison.
* Have no predefined methods.

Resources:

* <https://www.includehelp.com/scala/what-is-the-difference-between-scala-s-case-class-and-class.aspx#:~:text=Points%20of%20difference%20between%20Case%20Class%20and%20Class%20in%20Scala&text=The%20case%20class%20is%20defined,(syntax%20for%20defining%20class)>.
* <https://docs.scala-lang.org/tour/case-classes.html>

Observations:

* Basic code snippets for both classes were created for this question. Since Scala is a new language for me, I decided to create these code snippets so it could be easily understood. besides being initiated differently, we can see that the regular class “Regular” uses the keyword “new”. The differences above might be describing more details about these 2 classes, however I wanted to create basic examples for both.

A screenshot of a computer program

Description automatically generated with low confidence

1. Question 5

Resources (Please check Scala Worksheet Resources for more detail):

* <https://stackoverflow.com/questions/40526231/how-to-write-the-import-scala-io-source-import-java-io-libraries-in-sbt>
* <https://alvinalexander.com/scala/how-to-open-read-text-files-in-scala-cookbook-examples/>
* <https://www.oreilly.com/library/view/scala-cookbook/9781449340292/ch12s02.html>
* <https://stackoverflow.com/questions/34613697/add-a-new-line-of-text-to-an-existing-file-in-scala>
* <https://stackoverflow.com/questions/41953388/java-split-and-trim-in-one-shot>
* <https://www.scala-lang.org/api/2.13.5/scala/io/Source.html>
* <https://alvinalexander.com/scala/scala-split-string-example/>
* <https://stackoverflow.com/questions/26943460/how-to-create-map-for-each-line-based-on-the-column-using-scala>
* <https://stackoverflow.com/questions/19165977/how-does-scalas-groupby-identity-work>
* <https://alvinalexander.com/scala/scala-split-string-example/>

Observations:

* Code was creating using a mixture of Scala, O’Reilly, Stack Overflow documentation as well as internet recourses. We simply read the text file, which is located on my Desktop, and print line. It follows by adding the new row using “PrintWriter()” and 2 classes which are the “Class\_name” (Math repeats itself 6 times) and “Total\_Student” (get the total number of students in all courses -> 13).

A screenshot of a computer screen

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with medium confidence

Text file before/after adding the new row:

A screen shot of a computer

Description automatically generated with medium confidence

A screen shot of a computer

Description automatically generated with medium confidence

Closer look to both classes:

A screen shot of a computer

Description automatically generated with medium confidence

* .map(line => line.split([\\s+)](file://s+))) was used to transform each line. Splitting line into array of strings with [\\s](file://s)+ (split string based on a regular expression. Splitting strings on whitespace characters).

Code:

// Question 1

class Items {

private var x: String = \_ // \_ default value defined as null, 0, False

def get\_item\_from\_user(input: String): Unit = {

this.x = input // "this" key is used to fetch the variable of the class

}

def print\_String(): Unit = {

println(this.x.toUpperCase) // toUpperCase method

}

}

val x = new Items() // referring to class Items

x.get\_item\_from\_user("I had to read a lot for this Homework!")

x.print\_String()

/ Question 2

class Student(var student\_name: String, var marks: Int)

val name = new Student("Vincent De Leon", 95) //referring to class Student

// String interpolation

println(s"Student Name and Marks: ${name.student\_name}, ${name.marks}")

name.student\_name = "Carolina Licona"

name.marks = 99

// String Interpolation

println(s"Modified Student Name and Marks: ${name.student\_name}, ${name.marks}")

// Question 3

class Operations {

def addInt(a: Int, b:Int) : Int = { a + b }

def subInt(a: Int, b:Int) : Int = { a - b }

def mltpInt(a: Int, b:Int) : Int = { a \* b }

}

// Main method to execute the above 3 functions

// This Main runs in scala class not worksheet

object Main {

def main(args: Array[String]) : Unit = {

val operations = new Operations()

val add = operations.addInt(70, 5)

val sub = operations.subInt(70, 5)

val mltp = operations.mltpInt(70, 5)

println(add)

println(sub)

println(mltp)

}

}

// Question 4

class Private {

private var x: Int = \_

def display(): Unit = {

x = 5

println(x)

}

}

class Protected {

protected var y: Int = \_

def display(): Unit = {

y = 5

println(y)

}

}

// Using protected modifier with New1

class New1 extends Protected {

def display1(): Unit = {

y = 9

println(y)

}

}

class Public {

var z: Int = \_

}

var e = new Private()

e.display()

var e1 = new New1()

e1.display()

e1.display1()

var e2 = new Public()

e2.z = 222

println(e2.z)

// Question 5

// Differences between these 2 classes are within PDF File

// Case class

case class Case(x: String, age: Int) {

def result1(): Unit = {

println(s"Hey, my name is $x and I am $age years old.")

}

}

val name = Case("Carolina", 27)

name.result1()

// Regular Class

class Regular(val x: String, var age: Int) {

def result1(): Unit = {

println(s"Hey, my name is $x and I am $age years old.")

}

}

val name = new Regular("Carolina", 27)

name.result1()

// Question 6

import scala.io.Source

import java.io.\_

val source = Source.fromFile("Users/deleonv/Desktop/Student\_Class.txt")

for (line <- source.getLines) {

println(line)

}

source.close()

// Adding new row. Run this just 1 time to avoid overwriting.

val writer = new PrintWriter(new FileOutputStream(new File("Users/deleonv/Desktop/Student\_Class.txt"), true))

writer.println("M physics")

writer.close()

// If you want to read it again, comment the above code (Adding new row)

// and the code from Part A (reading txt file), then run this:

//val source = Source.fromFile("Users/deleonv/Desktop/Student\_Class.txt")

//for (line <- source.getLines) {

//println(line)

//}

//source.close()

// You will be able to see the new row within the Text File.

class Class\_name {

def max\_class(txt\_file: String): (String, Int) = {

val source = Source.fromFile(txt\_file)

val class\_list = source.getLines().drop(1).map(line => line.split("\\s+")).filter(\_.length > 1).map(\_(1))

val student\_class = class\_list.toList

// Find the class with the most students

val student\_count = student\_class.groupBy(identity).view.mapValues(\_.size).maxBy(\_.\_2)

source.close()

student\_count

}

}

val class\_name = new Class\_name()

val results\_c = class\_name.max\_class("Users/deleonv/Desktop/Student\_Class.txt")

println(s"The class with most students is (class, # students): $results\_c")

class Total\_Student {

def max\_students(txt\_file: String): Int = {

val source = Source.fromFile(txt\_file)

val class\_list = source.getLines().drop(1).map(line => line.split("\\s+")).filter(\_.length > 1).map(\_(1))

val student\_class = class\_list.toList

// Find the total number of students in all courses

val students = student\_class.size

source.close()

students

}

}

val total\_students = new Total\_Student()

val results\_d = total\_students.max\_students("Users/deleonv/Desktop/Student\_Class.txt")

println(s"The total number of students (counting all courses): $results\_d")