**Scala Basics**

**1. What is 2 to the power of 5 ?**

math.pow(2, 5)

res1: Double = 32.0

**2. What is the remainder of 180 divided by 7?**

180 % 7

res3: Int = 5

**3. Given variable pet\_name = "Sammy", use string interpolation to print out "My dog's name is Sammy."**

val name = "Sammy"

name: String = Sammy

var statement = "My dog's name is " + name

statement: String = My dog's name is Sammy

**4. Use Scala to find out if the letter sequence "xyz" is contained in: "sadfjshfjyuyxyzjkfuidkjklhasyysdfk"**

val key = "xyz"

val target = "sadfjshfjyuyxyzjkfuidkjklhasyysdfk"

target.contains(key)

res4: Boolean = true

**5. What is the difference between a value and a variable?**

A **variable** is mutable, while a **value** is immutable in Scala. For instance,

val myVal = 4

myVal = 5

<console>:25: error: reassignment to val

myVal = 5

^

But, using variables, this works fine:

var myVar = 4

myVar = 5

myVar: Int = 5

**6. Given the tuple (1,2,3,(4,5,6)) retrieve the number 6.**

val myTuple = (1,2,3,(4,5,6))

myTuple.\_4.\_3

res7: Int = 6

**Collections**

**1. Can you figure out what method you can use to find out if the list List(1,2,3,4,5) contains the number 3?**

val myList = List(1,2,3,4)

myList.contains(3)

Boolean = true

**2. How can you add all the elements of the previous list?**

myList.sum

Int = 10

**3. Create an Array of all the odd numbers from 0 to 15**

Array.range(1, 15, 2)

Array[Int] = Array(1, 3, 5, 7, 9, 11, 13)

**4. What are the unique elements in the list: List(2,3,1,4,5,6,6,1,2)?**

List(2,3,1,4,5,6,6,1,2).toSet

scala.collection.immutable.Set[Int] = Set(5, 1, 6, 2, 3, 4)

**5. Create a mutable map that maps together Names to Ages. It should have the following key value pairs:**

Sammy, 3

Frankie, 7

John, 45

val myMap = collection.mutable.Map(("Sammy", 3), ("Frankie", 7), ("John", 45))

myMap: scala.collection.mutable.Map[String,Int] = Map(Sammy -> 3, Frankie -> 7, John -> 45)

**5a. Print out all the keys**

myMap.keys

Iterable[String] = Set(Sammy, Frankie, John)

**5b. Add the key value pair ("Mike",27):**

myMap += ("Mike" -> 27)

myMap.type = Map(Sammy -> 3, Mike -> 27, Frankie -> 7, John -> 45)

**Functions**

**1. Check for single even: write a function that takes in an integer and returns a Boolean indicating whether or not it is even.**

def checkForSingleEven(target: Int): Boolean = {

return (target % 2 == 0)

}

**2. Check for Evens in a List: Write a function that returns True if there is an event number inside of a List, otherwise, return False.**

def checkForEvensInList(numbers: List[Int]): Boolean = {

for (num <- numbers){

if (num % 2 > 0){

return false

}

}

return true

}

**3. Lucky Number Seven: Take in a list of integers and calculate their sum. However, sevens are lucky and they should be counted twice, meaning their value is 14 for the sum. Assume the list isn't empty.**

def lucky(nums: List[Int]): Int = {

var output = 0

for (num <- nums){

if (num == 7){

output = output + 14

} else{

output = output + num

}

}

return output

}