### Swift Programming

**Swift Exam, Duration: 180 minutes**

# Submission:

**Name (Ad on Learning Hub): Yoshiki Sashiyama**

**Student ID: 20211453**

* Please read each questions and create the corresponding swift files and add them to the submission folder and at the end send it to me (Alireza) on Slack.
* The exam is out of 105 points
* There are 5 mandatory question and 1 optional (Question 6)
* Based on how you think you have done each problem you decide how much point to assign to that problem. To this end complete this table.
  + Note: Pick one the given points for each problem. (Make sure the total is 105)
  + Note: Make sure the total selected point is equal to 105.
  + Write done the point you pick for the problem next to the problem

|  |  |
| --- | --- |
| Problem | Point |
| Problem1 [15pts] | 15 |
| Problem2 [20pts] | 20 |
| Problem3 [20pts] | 20 |
| Problem4 [20pts] | 20 |
| Problem5 [25pts] | 25 |
| Problem4 [5pts] | 5 points |
| Total | 105 |

* Swift Documentation can be found here: <https://developer.apple.com/documentation/swift/>
* This is an individual exam please do not communicate with anyone during the exam.
* This is open book meaning that you have access to your Swift source code for your assignments and also Swift developer documentation.

# Problem1

Define an Enum, called WeekDay with the following cases:

* Monday
* Tuesday
* Wednesday
* Thursday
* Friday
* Saturday
* Sunday

Add a computed property to the enum which returns the number of day in the week as following:

|  |  |
| --- | --- |
| **Day of week** | **Number of the day** |
| Monday | 0 |
| Tuesday | 1 |
| Wednesday | 2 |
| Thursday | 3 |
| Friday | 4 |
| Saturday | 5 |
| Sunday | 6 |

**Submission:**

Insert the **snapshot** of the testing your code here in this file.Text

Description automatically generated

Upload the **problem3.swift** file to the submission folder.

# Problem2

The following snippet of code is given:

import Foundation

class Student{

private var age = 10

private var email = ""

private var takenCourses = 6

private var gpa = 68

init(\_ age: Int, \_ email: String, \_ takenCourses: Int, \_ gpa: Int){

self.age = age

self.email = email

self.takenCourses = takenCourses

self.gpa = gpa

}

}

class Search{

static func searchForStudents(list students: [Student], with condition: (Student)->Bool, their specification: (Student)->String,

action perform: (String)->Void){

for student in students{

if condition(student){

let spec = specification(student)

perform(spec)

}

}

}

}

let st1 = Student(23, "a.gmail.com", 5, 78)

let st2 = Student(22, "b.gmail.com", 8, 72)

let st3 = Student(19, "c.gmail.com", 7, 63)

let st4 = Student(25, "d.gmail.com", 6, 81)

let st5 = Student(24, "e.gmail.com", 4, 66)

let st6 = Student(22, "f.gmail.com", 7, 81)

var students:[Student] = []

students.append(st1)

students.append(st2)

students.append(st3)

students.append(st4)

students.append(st5)

students.append(st6)

Write code and insert the snapshots of the results for the following questions:

1. Call the searchForStudents method of the class Search to print the email address of the students who are older than 22
2. Call the searchForStudents method of the class Search to print the email address of the students who are between 20 and 23
3. Call the searchForStudents method of the class Search to print the email address and gpa of the students who have taken more than 5 courses
4. Call the searchForStudents method of the class Search to print the email address, gpa, age and number of courses all students in the list.

**Note:** You are not supposed to change the method searchForStudents

**Note:** You are allowed to add methods to the class Students if needed. But you are not allowed to change the original definition of the class Student.

**Note:** If you add any method to the class Student, please include the updated swift file.

**Submission:**

* Insert the **snapshot** of the testing your code here in this file.

A picture containing text

Description automatically generated

* Upload the **problem2.swift** file to the submission folder.

# Problem3

For each of the following example, design a class or structure (or both if needed) and identify which states (instance/class variables) each of the classes/structures have and create an instance from each class/structure with arbitrary values:

* A class/struct to show a mathematical Ellipse. A circle in mathematics is shown using the following formula:

aX^2 + bY^2 = 6

for example this is an ellipse: 5X^2+5Y^2 = 10

* A Directory could include files and other fire directories. For instance, in Directory IOSProject you can have the following:

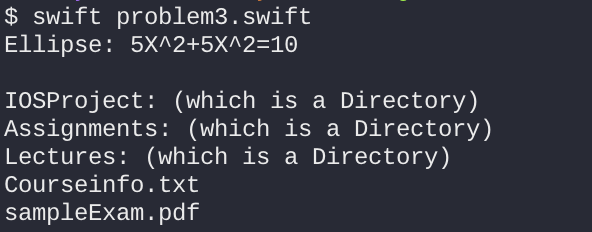
Assignments: (which is a Directory)

Lectures: (which is a Directory)

Courseinfo.txt

sampleExam.pdf

**Submission:**

* Insert the **snapshot** of the testing your code here in this file.
* Upload the **problem3.swift** file to the submission folder.

# Problem4

Please look at the following code.

* You are not supposed to make any change to the matchFind method.
* Any other change is allowed.
* Create one instance from class MatchFinder.
* Call matchFind method with the following input:
  + list: [“Vancouver”, “Toronto”, “calgary”, “Edmonton”, “surrey”, “victoria”]
  + test: a closure that returns true if number of vowel in a word is greater or equal to 3. For instance it returns true for Toronto but fals for surrey.
* For any match found by the matchFind method, the MatchFidner class somehow needs to notify the MatchListener class. The MatchListener will keep track of how match was found by the matchFind method.
* When the matchFind method terminates, the MatchListener class is supposed to print the total number of matches found by the matchFind method.
* Please complete the class MatchListener
* Make any changes to the MatchFinder class as needed.
* **You cannot change the matchFind method**

import Foundation

class MatchFinder{

var foundCounter: Int

init(){

self.foundCounter = 0

}

//This method is not supposed to change

func matchFind(\_ list: [String], \_ test:(String)->Bool){

for item in list{

if test(item){

foundCounter = foundCounter + 1

}

}

}

}

class MatchListener {

}

**Submission:**

* Insert the **snapshot** of the testing your code here in this file.
* Upload the **problem4.swift** file to the submission folder.

# Problem5

The following picture shows a suitcase containing several items. Each item in the suitcase has a weight shown the following table:

|  |  |
| --- | --- |
| Item | Weight (germs) |
| key | 50 |
| battery | 40 |
| watch | 100 |
| ring | 30 |
| iPhone | 300 |
| Cup | 250 |
| Notebook | 150 |

We need to know the following information. Develop a Swift program with required classes/structures, methods and properties to implement the following use cases:

* The total weight of items in the suitcase
* The name of the item with the highest weight in the suitcase
* The list of items sorted based on their weight ascendingly.
  + For sorting use the *sorted* method in Swift.
  + Note: Sorted(by: ) : https://developer.apple.com/documentation/swift/array/2296815-sorted

**Submission:**

* Insert the **snapshot** of the testing your code here in this file.Text

  Description automatically generated
* Upload the **problem5.swift** file to the submission folder.

A picture containing diagram

Description automatically generated

**Problem6 (Optional) – Free Bonus (5 points):**

There is no wrong answer to this question.

Your feedback is important to me.

You will get 5 points bonuses if you answer this question regardless of your comment or feedback.

Open question6 package and enter your answer there in the feedback file:

1. What have you liked about this course?
   1. Yes, but I would like to do more difficult things.
2. What have you disliked about this course so far?
   1. I always finished assignments fast so there were some boring times in this course so far.
3. What would be one or more thing you would do to improve the quality of the course and course delivery?
   1. I’m not sure, sorry.
4. 4-Did you find this exam Hard / Easy / Fair?
   1. Little bit easy.

**Good Luck ☺**