Brian’s Study Guide notes
CPSC-223W Swift
Midterm #2

All POGIL Activities

Thursday, November 5, 2020

Thu Oct 1, 2020	Assignment Exam 1	due by 3:20pm
Fri Oct 2, 2020	Assignment Lab exercise 5: Collections and Iterating through collections	due by 11:59pm
Tue Oct 6, 2020	Assignment POGIL Activity 7.1: Optionals	due by 11:59pm
	Assignment POGIL Activity 7.2: Type casting and Inspection	due by 11:59pm
Mon Oct 12, 2020	Assignment Lab exercise 6: Optionals	due by 11:59pm
Tue Oct 13, 2020	Assignment Quiz 5: Optionals, Type casting, and Inspection	due by 1:35pm
Fri Oct 16, 2020	Assignment POGIL Activity 8.1: Guard	due by 11:59pm
	Assignment POGIL Activity 8.2: Enumeration	due by 11:59pm
Mon Oct 19, 2020	Assignment Project Check-in 1	due by 11:59pm
Tue Oct 20, 2020	Assignment Quiz 6: Guards and Enumeration	due by 1:35pm
	Assignment Lab exercise 7: Guard and Enumeration	due by 11:59pm
	Assignment Lab exercise 8: Segues and Navigation Controllers	due by 11:59pm
	Assignment POGIL Activity 9.1: Protocols	due by 11:59pm
Thu Oct 22, 2020	Assignment POGIL Activity 9.2: Saving data	due by 11:59pm
Tue Oct 27, 2020	Assignment Quiz 6: Protocols and Saving data	due by 1:35pm
	Assignment Lab exercise 9: Protocols and Saving data	due by 11:59pm
	Assignment POGIL Activity 10.1: Closures	due by 11:59pm
Thu Oct 29, 2020	Assignment Project Check-in 2	due by 11:59pm
Tue Nov 3, 2020	Assignment Quiz 7: Closures	due by 1:35pm
	Assignment Lab exercise 10: Closures	due by 11:59pm
	Assignment POGIL Activity 11.1: Extensions	due by 11:59pm
Thu Nov 5, 2020	Assignment Exam 2	due by 3:20pm

Brian's Study Guide notes
CPSC-223W Swift
Midterm #2
All POGIL Activities
Thursday, November 5, 2020

POGIL Activity 7.1: Optionals

In this activity we will learn about optionals. Optionals provide a way for us to elegantly deal with nil values.

Please fill in the roles for each member of your team. Take a look at the description of each role to see its responsibilities. In case there are only three people in the group, please assign the same person to the **Presenter** and **Reflector** role. It is a good idea to select roles that you have not recently taken.

Team name: ______________________________________ Date: _______________

Role	**Team Member Name**
Manager. Keeps track of time and makes sure everyone contributes appropriately.	
Presenter. Talks to the facilitator and other teams.	
Reflector. Considers how the team could work and learn more effectively.	
Recorder. Records all answers and questions and makes the necessary submission.	

Brian's Study Guide notes
CPSC-223W Swift
Midterm #2
All POGIL Activities
Thursday, November 5, 2020

Part 1. Explore (5 min) Start time: ______

1. Your group will be assigned one of the following topics: (1) Functions and optionals, (2) Failable initializers, and (3) Optional Chaining. Read the corresponding section in Section 3.1 of the ADS book and discuss as a group.

Part 2. Invent (15 min) Start time: ______

2. What is the problem/issue/goal that your assigned topic addresses?

Functions and optionals
Allows functions to accept nil values

Failable initializers
Allows behavior where instance initialization returns nil if it can't be created

Optional Chaining
Reduce code you need to write for handling composed optional properties

3. Describe the syntax for using the feature.

Functions and optionals
Add a ? after the data type in a function parameter

Failable initializers
Use init? and return nil when the instance cannot be created

Optional Chaining
Same as the usual dot notation, but place a ? after the property name.

4. Provide an example that uses the feature.

Brian's Study Guide notes
CPSC-223W Swift
Midterm #2

All POGIL Activities

Thursday, November 5, 2020

Functions and optionals

```
func printFullName(firstName: String, middleName: String?,
lastName: String) {
    print("First name: \(firstName)")
      if let middle = middleName {
        print("Middle name: \(middle)")
      }
    print("Last name: \(lastName)")
}

printFullName(firstName: "Tuffy", middleName: "nil", lastName:
"Titan")
```

Failable initializers

```
struct Toddler {
  var name: String
  var monthsOld: Int

  init?(name: String, monthsOld: Int) {
    if monthsOld < 12 || monthsOld > 36 {
      return nil
    } else {
      self.name = name
      self.monthsOld = monthsOld
    }
  }
}

let possibleToddler = Toddler(name: "Joanna", monthsOld: 14)
if let toddler = possibleToddler {
  print("\(toddler.name) is \(toddler.monthsOld) months old")
} else {
  print("The age you specified for the toddler is not between
1 and 3 yrs of age")
}
```

Optional Chaining

```
struct Person {
  var age: Int
  var residence: Residence?
}

struct Residence {
  var address: Address?
}
```

```
struct Address {
  var buildingNumber: String
  var streetName: String
  var roomNumber: String?
}

var csAddress = Address(buildingNumber: "CS Building",
streetName: "800 N State College Blvd", roomNumber: "CS 101")
var csBuilding = Residence(address: csAddress)
var tuffy = Person(age: 63, residence: csBuilding)

if let room =
  tuffy.residence?.address?.roomNumber {
    print("They live in room number \(room).")
}
```

Part 3. Apply (30 min)

Start time: ______

5. Create the two structures described below:
    a. Website

    Create a Website structure that stores the title and URL of a website.

    b. SocialMediaAccount

    Create a SocialMediaAccount structure that stores the social media name (e.g., Facebook, Instagram, TIkTok), username, password, and website.

    The password should have 4 - 8 characters and the website can be optional. Design the initializer to take values for all the properties described above. However, if the password does not follow the requirements, return a nil object to indicate that the account cannot be created. Also make sure that you are able to create an account with a nil website. Hint: You can retrieve each character of a String, but that returns a Character object. You need to convert it to a String before converting it to an Int (see pg. 303 of the ADS).

    Create a WebSite and SocialMediaAccount instance. Provide any values you prefer. If the password is invalid, display "Invalid Password." Display the social media name, username, and website link (URL) of the account. Use optional chaining to display the website link to simplify your code.

Use Swift Playground to implement your solution. Run the program and take a screenshot of the code and output. Place the screenshot below.

```
struct WebSite {
    var title: String
    var url: String
}

struct SocialMediaAccount {
    var socialMediaName: String
    var username: String
    var password: String
    var website: WebSite?

    init?(socialMediaName: String, username: String,
          password: String, website: WebSite?) {
        if password.count < 4 || password.count > 8 {
            return nil
        }
        var hasNumber = false
        for char in password {
            if let _ = Int(String(char)) {
                hasNumber = true
            }
        }
        if !hasNumber {
            return nil
        }
        self.socialMediaName = socialMediaName
        self.username = username
        self.password = password
        self.website = website
    }
}

var tuffyWebSite = WebSite(title: "Home page",
                           url: "https://fullerton.edu")
if let account = SocialMediaAccount(
                   socialMediaName: "Facebook",
                   username: "tuffy", password: "Titan123",
                   website: tuffyWebSite) {
    print(account.socialMediaName)
    print(account.username)
    print(account.password)
    if let site = account.website?.url {
        print(site)
    }
} else {
    print("Invalid password.")
}
```

Reflector questions

1. What was the most useful thing you learned during this session?

2. What did the team do well?

3. What were the challenges that the team encountered?

4. What do you suggest the team do in the next meeting to do better?

5. Rate your team according to the rubric below

Self-rating:

Criteria	Score
Answered all problems in the worksheet	10
Partially answered the problems in the worksheet	8
Did not answer the worksheet	0

POGIL Activity 7.2: Type casting and Inspection

In this activity we will learn about type casting and inspection. These concepts facilitate the use of polymorphism. That is, we can create variables or collections that are able to store different kinds of objects and access methods depending on the type of object.

Please fill in the roles for each member of your team. Take a look at the description of each role to see its responsibilities. In case there are only three people in the group, please assign the same person to the **Presenter** and **Reflector** role. It is a good idea to select roles that you have not recently taken.

Team name: ______________________________   Date: ____________

Role	Team Member Name
Manager. Keeps track of time and makes sure everyone contributes appropriately.	
Presenter. Talks to the facilitator and other teams.	
Reflector. Considers how the team could work and learn more effectively.	
Recorder. Records all answers and questions and makes the necessary submission.	

Brian’s Study Guide notes
CPSC-223W Swift
Midterm #2
All POGIL Activities
Thursday, November 5, 2020

Part 1. Explore (10 min)   Start time: ______

1. Trace the code below. Identify the error and explain it. Feel free to run the code on Swift playground.

```
class Animal {
  var name: String
  init(name: String) {
    self.name = name
  }
  func speak() {
    print("\(name): ...")
  }
}

class Dog : Animal {
  override init(name: String) {
    super.init(name: name)
  }
  override func speak() {
    print("\(name): woof!")
  }

  func fetch() {
    print("\(name) fetches ....ʕ◦□◦ˀ")
  }
}

class Cat : Animal {
  override init(name: String) {
    super.init(name: name)
  }
  override func speak() {
    print("\(name): meow!")
  }

  func scratch() {
    print("\(name) scratches ....□ↀ◡ↀ□")
  }
}

var garfield = Cat(name: "Felix")
garfield.speak()
garfield.scratch()
```

```
var pluto = Dog(name: "Pluto")
pluto.speak()
pluto.fetch()

print("==Animal==")
let animal: Animal = garfield
animal.speak()

animal.scratch()
```

What was the error?
Value of type Animal has no member 'scratch'

What caused the error?
The animal instance is an Animal type. The Animal type does not provide a scratch member function

Part 2. Invent (15 min)

Start time: ______

2. Based on our discussion, how can you type cast animal so that you can call garfield's scratch method?

```
if let cat = animal as? Cat {
    cat.scratch()
}
```

3. Complete the code below by creating a loop that iterates over the underTheSun array. If the instance is a Dog, call it's fetch method. If the instance is a Cat, call it's scratch method. Otherwise, use print to display the value of the instance.

```
var garfield = Cat(name: "Felix")
var pluto = Dog(name: "Pluto")

var underTheSun: [Any] = [25, garfield, pluto, "Go Titans!"]
```

```
// add code below
for anything in underTheSun {
  if let cat = anything as? Cat {
    cat.scratch()
  } else if let dog = anything as? Dog {
    dog.fetch()
  } else {
    print(anything)
  }
  print()
}
```

Part 3. Apply (15 min) Start time: ______

4. Complete the function below so that it calls the associated method according to the type of the parameter. Specifically, call the fetch method if it is a Dog type, call the scratch method if it is a Cat type, and print the instance otherwise.

```
func performAction(actor: Any) {
  // add code below
  if let cat = actor as? Cat {
    cat.scratch()
  } else if let dog = actor as? Dog {
    dog.fetch()
  } else {
    print(actor)
  }

}

var garfield = Cat(name: "Felix")
performAction(actor: 223)
performAction(actor: garfield)
```

Reflector questions

1. What was the most useful thing you learned during this session?

2. What did the team do well?

3. What were the challenges that the team encountered?

4. What do you suggest the team do in the next meeting to do better?

5. Rate your team according to the rubric below

Self-rating:

Criteria	**Score**
Answered all problems in the worksheet	10
Partially answered the problems in the worksheet	8

Did not answer the worksheet	0

POGIL Activity 8.1: Guard

In this activity we will learn about guards in Swift, which are special constructs that promote early exit. Using early exit is a best practice in software development that promotes readability.

Please fill in the roles for each member of your team. Take a look at the description of each role to see its responsibilities. In case there are only three people in the group, please assign the same person to the **Presenter** and **Reflector** role. It is a good idea to select roles that you have not recently taken.

Team name: ______________________________________ Date: _______________

Role	**Team Member Name**
Manager. Keeps track of time and makes sure everyone contributes appropriately.	
Presenter. Talks to the facilitator and other teams.	
Reflector. Considers how the team could work and learn more effectively.	
Recorder. Records all answers and questions and makes the necessary submission.	

Brian's Study Guide notes
CPSC-223W Swift
Midterm #2
All POGIL Activities
Thursday, November 5, 2020

Part 1. Explore (10 min) Start time: ______

1. Explain the similarities and differences between the two functions below.

Code 1

```
func isValid(password: String) -> Bool {
  if password.count > 0 && password.count < 9 {
    var hasNumber = false
    for char in password {
      if char.isNumber {
        hasNumber = true
      }
    }
    if hasNumber {
      var hasLetter = false
      for char in password {
        if char.isLetter {
          hasLetter = true
        }
      }
      if hasLetter {
        return true
      } else {
        return false
      }
    } else {
      return false
    }
  } else {
    return false
  }
}
```

Code 2

```
func isValid(password: String) -> Bool {
  if password.count < 1 || password.count > 8 {
    return false
  }
  var hasNumber = false
  for char in password {
    if char.isNumber {
      hasNumber = true
    }
  }
  if !hasNumber {
    return false;
  }
  var hasLetter = false
  for char in password {
    if char.isLetter {
      hasLetter = true
    }
  }
  if !hasLetter {
    return false;
  }

  return true;
}
```

Similarities
The code returns true when the password follows the requirement and false otherwise.

Differences
Code 2 exits right away as soon as a requirement is not satisfied

Part 2. Invent (5 min) Start time: ______

2. What are the advantages of using Code 2?

Code is more readable because it is not nested. You can read the code linearly without going back to the previously defined conditions.

Part 3. Apply (20 min) Start time: ______

3. Modify the code in #1 to use the guard statement

```
func isValid(password: String) -> Bool {
  guard password.count > 0 && password.count < 9 else {
    return false
  }
  var hasNumber = false
  for char in password {
    if char.isNumber {
      hasNumber = true
    }
  }
  guard hasNumber else {
    return false;
  }
  var hasLetter = false
  for char in password {
    if char.isLetter {
      hasLetter = true
    }
  }
  guard hasLetter else {
    return false;
  }

  return true;
}
```

Reflector questions

1. What was the most useful thing you learned during this session?

2. What did the team do well?

3. What were the challenges that the team encountered?

4. What do you suggest the team do in the next meeting to do better?

5. Rate your team according to the rubric below

Self-rating:

Criteria	**Score**
Answered all problems in the worksheet	10
Partially answered the problems in the worksheet	8
Did not answer the worksheet	0

POGIL Activity 8.2: Enumeration

In this activity we will learn to use Enumerations to make code readable. Specifically it allows us to define variables that can take a finite set of values.

Please fill in the roles for each member of your team. Take a look at the description of each role to see its responsibilities. In case there are only three people in the group, please assign the same person to the **Presenter** and **Reflector** role. It is a good idea to select roles that you have not recently taken.

Team name: ____ Date: _______________

Role	**Team Member Name**
Manager. Keeps track of time and makes sure everyone contributes appropriately.	
Presenter. Talks to the facilitator and other teams.	
Reflector. Considers how the team could work and learn more effectively.	
Recorder. Records all answers and questions and makes the necessary submission.	

Brian’s Study Guide notes
CPSC-223W Swift
Midterm #2
All POGIL Activities
Thursday, November 5, 2020

Part 1. Explore (10 min)   Start time: ______

Observe the code below while focusing on the role property of the User struct.

```
struct Post {
  var title: String
  var author: String
  var content: String
}

struct User {
  var username: String
  var role: String
  var feed : [Post]

  init(username: String, role: String) {
    self.username = username
    self.role = role
    self.feed = []
  }

  mutating func add(post: Post) {
    feed.append(post)
  }

  func displayPosts() {
    for post in feed {
      if role == "Admin" {
        print(post.title)
        print(post.author)
        print(post.content)
      } else if role == "Member" {
        print(post.title)
        print(post.content)
      } else if role == "Guest" {
        print(post.title)
      }
      print("\n")
    }
  }
}

var post1 = Post(title: "Faculty and Staff Receive Nearly $9
Million in Grants, Contracts in Q1", author:
"strategiccomm@fullerton.edu", content: "Faculty and staff
garnered close to $9 million in funding during the first
quarter of the university's 2020-21 fiscal year.")

var post2 = Post(title: "Students Fired Up to Develop
Miniature Satellite to Fight Wildfires", author:
"strategiccomm@fullerton.edu", content: "As wildfires continue
to cause destruction across California, a team of engineering
students is exploring a solution to fight the devastating
infernos by designing and building a miniaturized satellite
known as a CubeSat.")

var post3 = Post(title: "Faculty Showcase Expertise Through
Publications, Presentations and Awards", author:
"strategiccomm@fullerton.edu", content: "See a listing of the
recent publications, presentations and awards from Cal State
Fullerton faculty.")

var me = User(username: "Paul", role: "Admin")
me.add(post: post1)
me.add(post: post2)
me.add(post: post3)

me.displayPosts()
```

1. What are the potential issues with the current code design?

The role names should match the values exactly. Mistyping it, even with lower case letters, will result in incorrect code behavior. It is also difficult for developers to figure out the possible values for role. They will need to read the displayPosts function and other parts of the code that us

Part 2. Invent (5 min)

Start time: ______

2. Read unit 3.5 Enumeration in the ADS book.How can Enumeration potentially address the issues you identified in #1?

Enumeration can solve the problem because it provides a finite set of possible values. An enumeration can be used Instead of using strings, ints, or other values so that you can only choose values from the set. It is also easy to see possible values for code that uses enumeration, because you only need to look at the definition of the enumeration.

Part 3. Apply (20 min)

Start time: ______

3. Create an Enumeration and modify the code in #1 to use your Enumeration.

```
enum Role {
  case admin
  case member
  case guest
}

struct Post {
  var title: String
  var author: String
  var content: String
}

struct User {
  var username: String
  var role: Role
  var feed : [Post]

  init(username: String, role: Role) {
    self.username = username
    self.role = role
    self.feed = []
  }

  mutating func add(post: Post) {
    feed.append(post)
  }

  func displayPosts() {
    for post in feed {
      switch role {
```

```
    case .admin:
     print(post.title)
     print(post.author)
     print(post.content)
    case .member:
     print(post.title)
     print(post.content)
    case .guest:
     print(post.title)
   }
   print("\n")
  }
 }
}

var post1 = Post(title: "Faculty and Staff Receive Nearly $9
Million in Grants, Contracts in Q1", author:
"strategiccomm@fullerton.edu", content: "Faculty and staff
garnered close to $9 million in funding during the first
quarter of the university's 2020-21 fiscal year.")

var post2 = Post(title: "Students Fired Up to Develop
Miniature Satellite to Fight Wildfires", author:
"strategiccomm@fullerton.edu", content: "As wildfires continue
to cause destruction across California, a team of engineering
students is exploring a solution to fight the devastating
infernos by designing and building a miniaturized satellite
known as a CubeSat.")

var post3 = Post(title: "Faculty Showcase Expertise Through
Publications, Presentations and Awards", author:
"strategiccomm@fullerton.edu", content: "See a listing of the
recent publications, presentations and awards from Cal State
Fullerton faculty.")

var me = User(username: "Paul", role: .admin)
me.add(post: post1)
me.add(post: post2)
me.add(post: post3)

me.displayPosts()
```

Reflector questions

1. What was the most useful thing you learned during this session?

2. What did the team do well?

3. What were the challenges that the team encountered?

4. What do you suggest the team do in the next meeting to do better?

5. Rate your team according to the rubric below

Self-rating:

Criteria	**Score**
Answered all problems in the worksheet	10
Partially answered the problems in the worksheet	8

Did not answer the worksheet	0

POGIL Activity 9.1: Protocols

In this activity we will learn about protocols that allow us to require functionalities from classes. Protocols allow us to design code that is easier to adapt and scale.

Please fill in the roles for each member of your team. Take a look at the description of each role to see its responsibilities. In case there are only three people in the group, please assign the same person to the **Presenter** and **Reflector** role. It is a good idea to select roles that you have not recently taken.

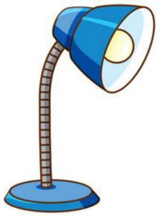
Team name: ____ Date: ________________

Role	Team Member Name
Manager. Keeps track of time and makes sure everyone contributes appropriately.	
Presenter. Talks to the facilitator and other teams.	
Reflector. Considers how the team could work and learn more effectively.	
Recorder. Records all answers and questions and makes the necessary submission.	

Brian’s Study Guide notes
CPSC-223W Swift
Midterm #2
All POGIL Activities
Thursday, November 5, 2020

Part 1. Explore (5 min) Start time: ______

Observe the images below

Port	Devices
	

1. Why is it possible to connect different devices to the same port?

> The devices follow a standard. For example, lamps, microwaves, and TVs have a standard-sized plug and use the same wattage. Mice, cameras, and keyboards have a USB that fits the standard USB port size and supports the USB protocol to send and

receive information.

Part 2. Invent (10 min) Start time: ______

2. Let's imagine that we are developing a new standard so that we can design a robot that uses different cooking appliances to cook food. What do you think are the common properties and functionalities that all cooking appliances share? (e.g., microwave, oven, toaster, air fryer).

They have a timer and are able to start the cooking process whose duration is based on the timer.

Part 3. Apply (25 min) Start time: ______

3. Design a CookingAppliance protocol and define properties and functions according to what you identified in #2. Make sure the protocol contains at least one function named start.

```
protocol CookingAppliance {
  var timer : Int {get set}
  func start()
}
```

4. Select one of the cooking appliances you considered in #2 and create a struct to represent it. Make sure it uses the CookingAppliance and CustomStringConvertible protocols. You can print values on the screen to simulate the cooking process.

```
struct AirFryer : CookingAppliance, CustomStringConvertible {
  var timer: Int
  var description: String {
    return "Air fryer with timer set to \(timer)."
  }
  init() {
    timer = 0
  }
  func start() {
    print("Air frying: ", terminator:"")
    for i in 1...timer {
      print("\(i) … ", terminator:"")
    }
    print(" Cooked!")
  }
}
```

5. Will your code work with the following struct? Explain why.

```
struct RoboChef {
  var name: String
  var appliance: CookingAppliance
  mutating func cook(food: String) {
    print("Cooking \(food)");
    appliance.timer = 15;
    appliance.start()
  }
}

// Call the constructor of your CookingAppliance beside the =
// to create an instance of your struct

let myAppliance =
var chef = RoboChef(name: "Gordon", appliance: myAppliance)
chef.cook(food: "potato")
```

Explanation

It will work if AirFryer has a timer property and a start function.

6. If anyone now or in the future decides to create a struct that uses the CookingAppliance protocol, will that new class also work with the RoboChef struct in #5? Why?

Just like the class we defined, it will also have the timer property and start function so it will also work with the RoboChef struct.

Reflector questions

1. What was the most useful thing you learned during this session?

2. What did the team do well?

3. What were the challenges that the team encountered?

4. What do you suggest the team do in the next meeting to do better?

5. Rate your team according to the rubric below

Brian's Study Guide notes
CPSC-223W Swift
Midterm #2

All POGIL Activities

Thursday, November 5, 2020

Self-rating:

Criteria	**Score**
Answered all problems in the worksheet	10
Partially answered the problems in the worksheet	8
Did not answer the worksheet	0

POGIL Activity 9.2: Saving data

In this activity we will learn how to encode, decode, store, and load data. You are expected to use your prior understanding of Swift programming constructs and practice understanding documentation.

Please fill in the roles for each member of your team. Take a look at the description of each role to see its responsibilities. In case there are only three people in the group, please assign the same person to the **Presenter** and **Reflector** role. It is a good idea to select roles that you have not recently taken.

Team name: ______________________________________ Date: _______________

Role	Team Member Name
Manager. Keeps track of time and makes sure everyone contributes appropriately.	
Presenter. Talks to the facilitator and other teams.	
Reflector. Considers how the team could work and learn more effectively.	
Recorder. Records all answers and questions and makes the necessary submission.	

Brian's Study Guide notes
CPSC-223W Swift
Midterm #2
All POGIL Activities
Thursday, November 5, 2020

Part 1. Explore (15 min) Start time: ______

Read pp. 659 - 666 in the App Development with Swift Book. Specifically, read from Encoding and Decoding with Codable to Saving an Array of Model Data. You will need to recall and apply some of the concepts we have discussed in class to understand some of the discussions.

Part 2. Invent (15 min) Start time: ______

1. Provide code that would use a PropertyListEncoder to encode data.

```
let newNote = Note(title: "Grocery run", text: "Pick up
mayonnaise, mustard, lettuce, tomato, and pickles.",
timestamp:
Date())
let propertyListEncoder = PropertyListEncoder()
if let encodedNote = try? propertyListEncoder.encode(newNote)
{
    print(encodedNote)
}
```

2. Provide code that would use a PropertyListDecoder to decode an encoded object.

```
let propertyListDecoder = PropertyListDecoder()
if let decodedNote = try? propertyListDecoder
  .decode(Note.self, from: encodedNote)  {
    print(decodedNote)
}
```

3. What is a sandbox and how is it relevant to files in Swift?

Swift uses a sandbox to manage files for security. File access is limited to the sandbox to avoid editing or deleting important files.

4. Provide code that would represent a file URL in the user's document directory (any file name will do).

```
let documentsDirectory =
FileManager.default.urls(for: .documentDirectory,
in: .userDomainMask).first!
let archiveURL =
documentsDirectory.appendingPathComponent("notes_test").
appendingPathExtension("plist")
```

5. Provide code that would write data into a file in the user's document directory (any file name will do).

```
try? encodedNote?.write(to: archiveURL, options:
.noFileProtection)
```

6. Provide code that would read data from a file in the user's document directory into a variable.

```
let propertyListDecoder = PropertyListDecoder()
if let retrievedNoteData = try? Data(contentsOf: archiveURL),
    let decodedNote = try?
propertyListDecoder.decode(Note.self, from: retrievedNoteData)
{
    print(decodedNote)
}
```

Part 3. Apply (15 min) Start time: ______

Consider the Contact struct below to answer the following questions.

```
struct Contact {
  var name: String
  var number: String
}
```

7. Write code that would store a Contact instance called myContact into a file called contacts.plist.

```
let newContact = Contact(name: "Tuffy", number:
"657-278-2011")

let documentsDirectory =
FileManager.default.urls(for: .documentDirectory,
in: .userDomainMask).first!
let archiveURL =
documentsDirectory.appendingPathComponent("contacts").
appendingPathExtension("plist")

let propertyListEncoder = PropertyListEncoder()
if let encodedContact = try?
propertyListEncoder.encode(newContact) {
  try? encodedContact?.write(to: archiveURL, options:
.noFileProtection)
}
```

8. Write code that would read a Contact object from contacts.plist and store it in a Contact instance called loadedContact.

```
var loadedContact: Contact

let propertyListDecoder = PropertyListDecoder()
let archiveURL =
documentsDirectory.appendingPathComponent("contacts").
appendingPathExtension("plist")

if let retrievedContact = try? Data(contentsOf: archiveURL),
    let decodedContact = try?
propertyListDecoder.decode(Contact.self, from:
retrievedContact) {
    loadedContact = decodedContact
}
```

Reflector questions

1. What was the most useful thing you learned during this session?

2. What did the team do well?

3. What were the challenges that the team encountered?

4. What do you suggest the team do in the next meeting to do better?

5. Rate your team according to the rubric below

Self-rating:

Criteria	**Score**
Answered all problems in the worksheet	10
Partially answered the problems in the worksheet	8
Did not answer the worksheet	0

Brian’s Study Guide notes
CPSC-223W Swift
Midterm #2
All POGIL Activities
Thursday, November 5, 2020

POGIL Activity 10.1: Closures

In this activity we will learn about closures. Closures allow us to define functions that we can store in variables together with other definitions that are available within its scope. Closures enable commonly used designs such as function callbacks, retrieving appropriate implementations, and the strategy design pattern.

Please fill in the roles for each member of your team. Take a look at the description of each role to see its responsibilities. In case there are only three people in the group, please assign the same person to the **Presenter** and **Reflector** role. It is a good idea to select roles that you have not recently taken.

Team name: ______________________________ Date: ____________

Role	**Team Member Name**
Manager. Keeps track of time and makes sure everyone contributes appropriately.	
Presenter. Talks to the facilitator and other teams.	
Reflector. Considers how the team could work and learn more effectively.	
Recorder. Records all answers and questions and makes the necessary submission.	

Brian's Study Guide notes
CPSC-223W Swift
Midterm #2
All POGIL Activities
Thursday, November 5, 2020

Part 1. Explore (5 min) Start time: ______

You are assigned to read and understand one of the following functions that use closures from the ADS book: map (pp. 789 - 791), filter (pp. 791 - 793), reduce (pp. 793 - 794). Make sure to understand the concept as you will be asked to explain it in class.

Part 2. Invent (5 min) Start time: ______

1. Provide sample code for using the function you were assigned in part 1.

Map

```
let names = ["Johnny", "Nellie", "Aaron", "Rachel"]

// Creates a new array of full names by adding "Smith" to each
// first name
let fullNames = names.map { (name) -> String in
    return name + " Smith"
}
```

Filter

```
let numbers = [4, 8, 15, 16, 23, 42]
let numbersLessThan20 = numbers.filter { (number) -> Bool in
    return number < 20
}

print(numbersLessThan20)
```

Reduce

```
let numbers = [8, 6, 7, 5, 3, 0, 9]
let total = numbers.reduce(0) { (currentTotal, newValue) ->
Int in
    return currentTotal + newValue
}
```

Part 3. Apply (30 min) Start time: ______

2. Let's assume that your grocery is having a sale and all products that are \$0.50 and below are free! A Product struct is provided for you below. You must use the map, filter, and reduce functions to write a computeSalePrice function. The function accepts an array of Products and returns a Double value. It should perform the following steps:
    a. Extract the prices of each element in the list of products
    b. Select only prices that are \$0.50
    c. Add all prices
    d. Return the total price

```
struct Product {
  var name: String
  var price: Double
  var quantity: Int
  var totalPrice: Double {
    Double(quantity) * price
  }
}
```

```
func computeSalePrice(products: [Product]) -> Double {
    let productPrices = products.map {
        (product) -> Double in
        return product.totalPrice
    }
    let filteredPrices = productPrices.filter {
        (price) -> Bool in
        return price > 0.5
    }
    let total = filteredPrices.reduce(0) {
        (currentTotal, newValue) -> Double in
        return currentTotal + newValue
    }
    return total
}
```

```
var milk = Product(name: "Milk", price: 3.99, quantity: 2)
var gum = Product(name: "Gum", price: 0.50, quantity: 1)
var roastedChicken = Product(name: "Roasted Chicken",
                             price: 7.99, quantity: 2)

var groceries = [milk, gum, roastedChicken]
let finalPrice = computeSalePrice(products: groceries)
print(finalPrice)
```

3. Create a displayProducts function that will display the contents of an array of Products. We will use closures to let developers control how to display the values on the screen (or some other form of output). The displayProducts function should accept two parameters, the array of Products with an omitted argument and a closure. The closure should accept three parameters, a String, a Double, and an Int. It should not return anything. The String parameter refers to the name of the product, the Double is the price and the Int is the quantity. The example below shows how your function might be used.

```
// place your code here
func displayProducts(products: [Product], displayFunction:
(String, Double, Int) -> Void) -> Void {
    for product in products {
        displayFunction(product.name, product.price,
                        product.quantity)
    }
}
```

```
// Sample usage of the displayProducts function

var products: [Product] = []
products.append(Product(name: "Milk", price: 3.99,
                        quantity: 1))
products.append(Product(name: "Gum", price: 0.50, quantity:
1))

displayProducts(products: products) {
  (name, price, quantity) -> Void in
  print("\(quantity) x \(name) : $\(price)")
}
```

Reflector questions

1. What was the most useful thing you learned during this session?

2. What did the team do well?

3. What were the challenges that the team encountered?

4. What do you suggest the team do in the next meeting to do better?

5. Rate your team according to the rubric below

Self-rating:

Criteria	**Score**
Answered all problems in the worksheet	10
Partially answered the problems in the worksheet	8

Did not answer the worksheet	0

POGIL Activity 11.1: Extensions

In this activity we will learn about extensions. Extensions allow us to add functionalities to existing classes, structures, enumerations, or protocols. They are also use to organize code and promote readability.

Please fill in the roles for each member of your team. Take a look at the description of each role to see its responsibilities. In case there are only three people in the group, please assign the same person to the **Presenter** and **Reflector** role. It is a good idea to select roles that you have not recently taken.

Team name: ____________________________________ Date: ______________

Role	**Team Member Name**
Manager. Keeps track of time and makes sure everyone contributes appropriately.	
Presenter. Talks to the facilitator and other teams.	
Reflector. Considers how the team could work and learn more effectively.	
Recorder. Records all answers and questions and makes the necessary submission.	

Brian's Study Guide notes
CPSC-223W Swift
Midterm #2
All POGIL Activities
Thursday, November 5, 2020

Part 1. Explore (10 min) Start time: ______

Read Lesson 5.2: Extensions of the App Development with Swift book (pp. 796 - 801). Pay particular attention to the purpose of extensions and their syntax.

Part 2. Invent (10 min) Start time: ______

1. What are the two things extensions are mostly used for? Describe each one.

Purpose 1 It is used to extend the functionality of an existing structure, class enumeration or protocol.
Purpose 2 It is used to organize code by placing them in a separate block to promote readability.

Part 3. Apply (15 min) Start time: ______

2. Extend the String structure to provide a titlecased() function. This modifies the String so that the first letter of each word is converted to uppercase. Let's assume that each word is separated by a space. For example, "hello world" becomes "Hello World".

```
extension String {
  func titlecased() -> String {
    let result = reduce("") {
      (runningString, newChar) -> String in
      var append: String = String(newChar)
      if runningString == "" || runningString.last == " " {
        append = newChar.uppercased()
      } else {
        append = newChar.lowercased()
      }
      return runningString + append
    }
    return result
  }
}
```

3. Refactor the SoccerGame struct by moving its methods into an extension.

```
struct SoccerGame {
  var homeTeam: String
  var visitingTeam: String
  var schedule: String

  var duration: Double
  var homeTeamScore: Int
  var visitingTeamScore: Int

  init(homeTeam: String, visitingTeam: String,
       schedule: String) {
    self.homeTeam = homeTeam
    self.visitingTeam = visitingTeam
    self.schedule = schedule
    homeTeamScore = 0
    visitingTeamScore = 0
    duration = 90
  }

  func showGameInfo() -> Void {
    print("\(homeTeam) vs. \(visitingTeam)")
    print("\(schedule)")
  }
  func showScore() -> Void {
    print("\(homeTeam) \(homeTeamScore) - \(visitingTeamScore)
          \(visitingTeam)")
    print("\(duration)")
  }

  func showWinner() -> Void {
    guard duration == 0 else {
      print("Match between \(homeTeam) and \(visitingTeam) has
            not yet started.")
      return
    }
    if homeTeamScore > visitingTeamScore {
      print("\(homeTeam) wins over \(visitingTeam)")
      print("\(homeTeamScore) -  \(visitingTeamScore)")
    } else {
      print("\(visitingTeam) wins over \(homeTeam)")
      print("\(visitingTeamScore) -  \(homeTeamScore)")
    }
  }
}
```

Refactored version using extensions

```
struct SoccerGame {
  var homeTeam: String
  var visitingTeam: String
  var schedule: String

  var duration: Double
  var homeTeamScore: Int
  var visitingTeamScore: Int
}

extension SoccerGame {
  init(homeTeam: String, visitingTeam: String,
       schedule: String) {
    self.homeTeam = homeTeam
    self.visitingTeam = visitingTeam
    self.schedule = schedule
    homeTeamScore = 0
    visitingTeamScore = 0
    duration = 90
  }

  func showGameInfo() -> Void {
    print("\(homeTeam) vs. \(visitingTeam)")
    print("\(schedule)")
  }

  func showScore() -> Void {
    print("\(homeTeam) \(homeTeamScore) - \(visitingTeamScore)
          \(visitingTeam)")
    print("\(duration)")
  }

  func showWinner() -> Void {
    guard duration == 0 else {
      print("Match between \(homeTeam) and \(visitingTeam) has
            not yet started.")
      return
    }
    if homeTeamScore > visitingTeamScore {
      print("\(homeTeam) wins over \(visitingTeam)")
      print("\(homeTeamScore) -  \(visitingTeamScore)")
    } else {
      print("\(visitingTeam) wins over \(homeTeam)")
      print("\(visitingTeamScore) -  \(homeTeamScore)")
    }
  }
}
```

Reflector questions

1. What was the most useful thing you learned during this session?

2. What did the team do well?

3. What were the challenges that the team encountered?

4. What do you suggest the team do in the next meeting to do better?

5. Rate your team according to the rubric below

Self-rating:

Criteria	**Score**
Answered all problems in the worksheet	10
Partially answered the problems in the worksheet	8
Did not answer the worksheet	0