

CSE 4500 – Platform Computing
Professor Lawrence Orijuela
Homework 3 (Due TBD) – Learn Swift Exercises
(Accompanying Lecture Vids:

<https://www.youtube.com/watch?v=yFYJph-tO4Q>
<https://www.youtube.com/watch?v=x00pLRvxmmc>
<https://www.youtube.com/watch?v=EbCxPgXnXv8>
)

Instructions:

1. Download this assignment as an editable word doc from Canvas.
2. Write your name and student ID.
3. For written questions (if any), write your answers in the spaces provided.
Give yourself more space as needed.
4. Export this doc with your name and answers as a PDF and resubmit it to canvas.
 - a. Use this as your title: CSE4500_YourName_HW#
 - b. Example: CSE4500_BobSmith_HW#3

Name: Ali Becerra Jr. _____ Student ID #: 007937560 _____ Points: _____/30 _____

Complete all of the Swift activities listed below. These are also on the 4500_1_LearningSwift.pptx. Please watch the accompanying lecture video (links above) at the same time as you are doing the assignment. I provide tips and hints on how to do these problems.

There are 4 activities total. Please copy-paste your code and screenshots in the designated areas.

Activity #1

1. Coding:

1. Write a simple swift program that determines the surface area of a rectangular prism.
 - Write three variables called *length*, *width*, and *height* respectively.
 - Write the proper math formula to determine the combined surface area of the six sides.
 - Print the surface area to the user.

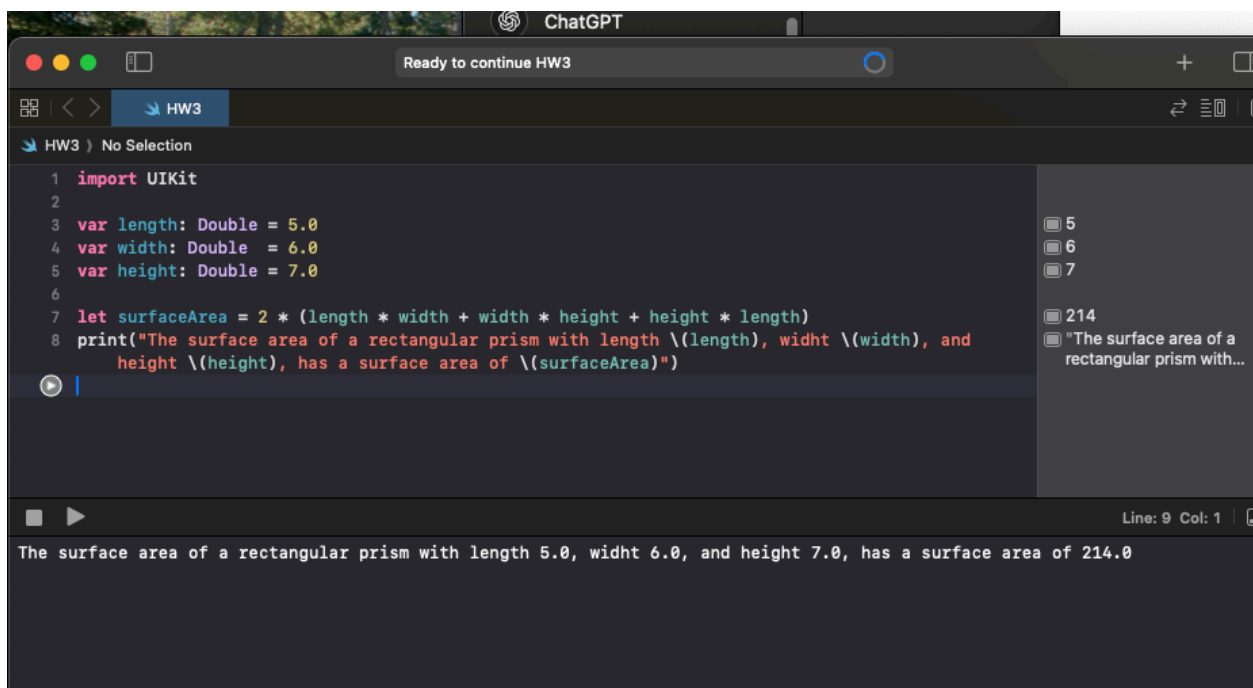
2. Deliverables:

- Copy-paste the code into the HW3 Assignment.
- Take a screenshot of each of the following test cases:
 1. Length = 5, Width = 6, Height = 7
 2. Length = 10, Width = 4, Height = 6

```
import UIKit
```

```
var length: Double = 5.0
var width: Double = 6.0
var height: Double = 7.0
```

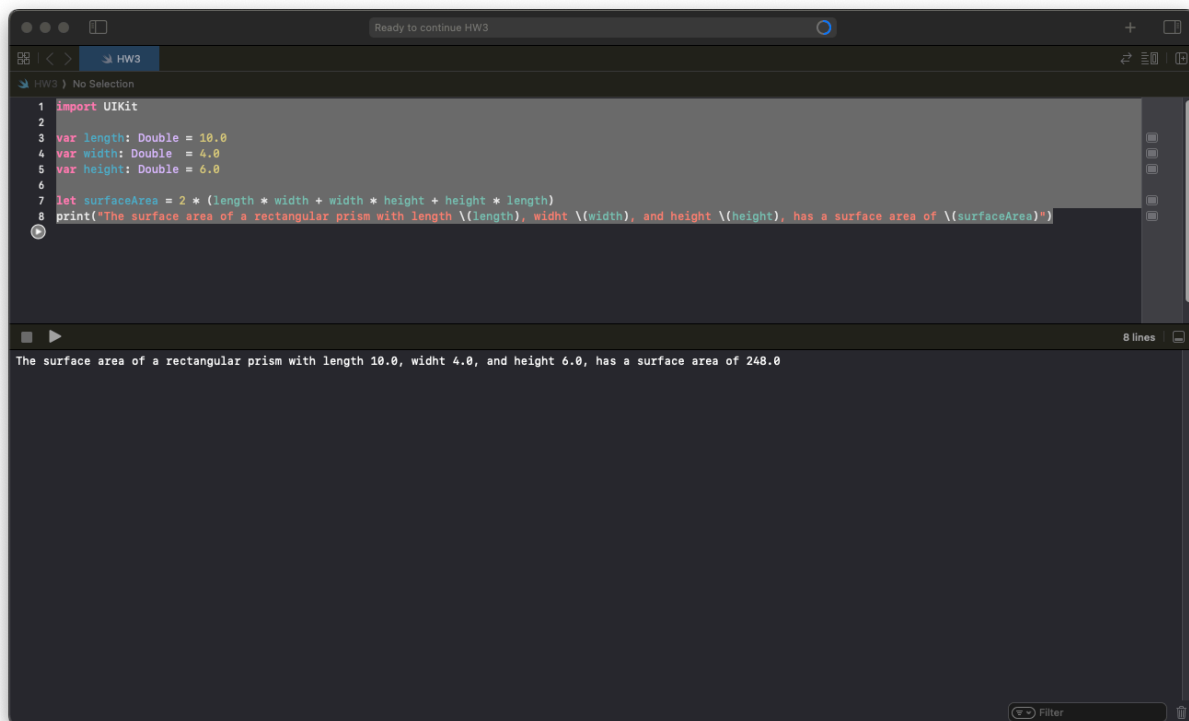
```
let surfaceArea = 2 * (length * width + width * height + height * length)
print("The surface area of a rectangular prism with length \(length), width \(width), and height \(height), has a surface area of \(surfaceArea)")
```



```
import UIKit
```

```
var length: Double = 10.0  
var width: Double = 4.0  
var height: Double = 6.0
```

```
let surfaceArea = 2 * (length * width + width * height + height  
* length)  
print("The surface area of a rectangular prism with length \  
(length), width \((width), and height \((height), has a surface  
area of \((surfaceArea)")
```



The screenshot shows a Swift Playground window titled "Ready to continue HW3". The code editor contains the following Swift code:

```
1 import UIKit  
2  
3 var length: Double = 10.0  
4 var width: Double = 4.0  
5 var height: Double = 6.0  
6  
7 let surfaceArea = 2 * (length * width + width * height + height * length)  
8 print("The surface area of a rectangular prism with length \((length), width \((width), and height \((height), has a surface area of \((surfaceArea)")
```

The output area at the bottom displays the result of the print statement:

```
The surface area of a rectangular prism with length 10.0, width 4.0, and height 6.0, has a surface area of 248.0
```

The output area also indicates "8 lines" of code were executed.

Activity #2

3. Write a simple Swift program that determines the type of a triangle from the length of its sides.
 - Create three variables called side1, side2, side3.
 - Write a series of if statements that will print either scaline, isosceles, equilateral, or not a triangle.
 1. Scaline triangles have 3 sides of differing lengths.
 2. Isosceles triangles have 2 sides equal, while one side is different.
 3. Equilateral triangles have all 3 sides the same.
 4. If any one side is larger than the other two combined, it is not a triangle.
4. Deliverables:
 - Copy-paste your code into the HW3 assignment:
 - Take a screenshot for each of the test cases:
 1. Side1 = 5, side2 = 6, side3 = 7. Expected Output: "Scalene triangle"
 2. Side1 = 11, side2 = 5, side3 = 5. Expected Output: "Not a triangle"

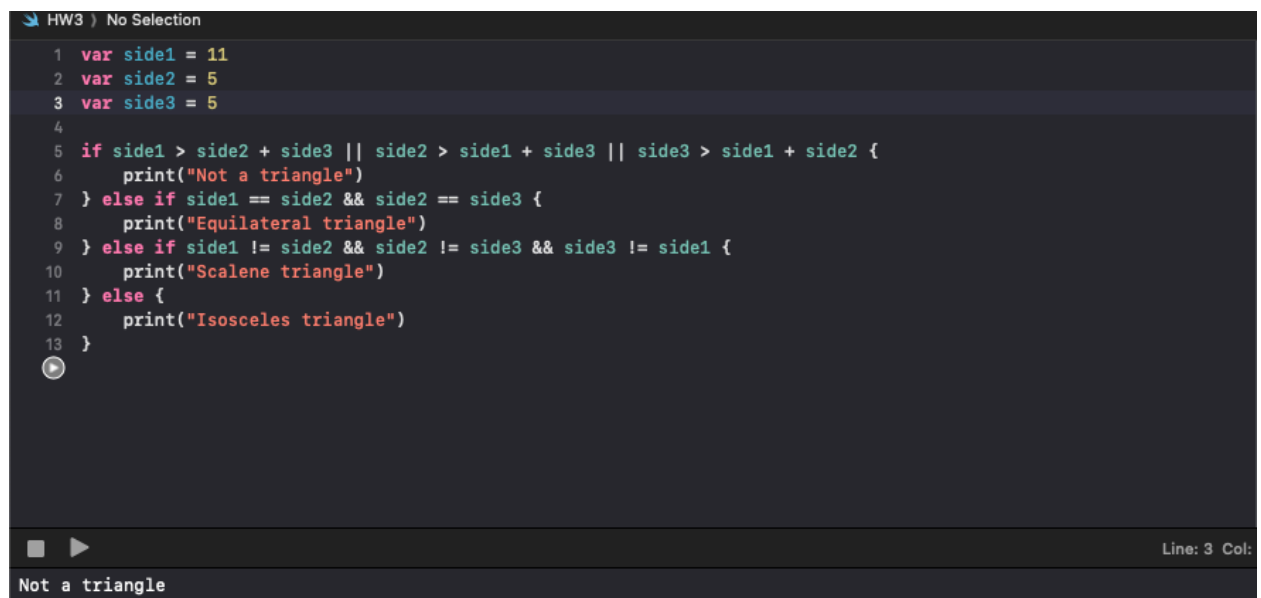
```
var side1 = 5
var side2 = 6
var side3 = 7
```

```
if side1 > side2 + side3 || side2 > side1 + side3 || side3 >
side1 + side2 {
    print("Not a triangle")
} else if side1 == side2 && side2 == side3 {
    print("Equilateral triangle")
} else if side1 != side2 && side2 != side3 && side3 != side1 {
    print("Scalene triangle")
} else {
    print("Isosceles triangle")
}
```



```
var side1 = 11
var side2 = 5
var side3 = 5
```

```
if side1 > side2 + side3 || side2 > side1 + side3 || side3 >
side1 + side2 {
    print("Not a triangle")
} else if side1 == side2 && side2 == side3 {
    print("Equilateral triangle")
} else if side1 != side2 && side2 != side3 && side3 != side1 {
    print("Scalene triangle")
} else {
    print("Isosceles triangle")
}
```



HW3) No Selection

```
1 var side1 = 11
2 var side2 = 5
3 var side3 = 5
4
5 if side1 > side2 + side3 || side2 > side1 + side3 || side3 > side1 + side2 {
6     print("Not a triangle")
7 } else if side1 == side2 && side2 == side3 {
8     print("Equilateral triangle")
9 } else if side1 != side2 && side2 != side3 && side3 != side1 {
10    print("Scalene triangle")
11 } else {
12    print("Isosceles triangle")
13 }
```

Line: 3 Col:

Not a triangle

Activity #3

5. Write a simple swift program finds the total cost of coins at a CoinStar:
 - Write a dictionary with the following key-value pairs that uses the name (string) of the coin as a key, and it's dollar value (double) as the value:
 1. Quarter, \$0.25
 2. Dime, \$0.20
 3. Nickel, \$0.05
 4. Penny, \$0.01
 - Create an array called *myChange* containing how many of each coin you have in your coin bank.
 - Write a series of functions that will use optional binding to reference the dictionary values by key and multiply them by each sequential item in your array and return the total dollar amount that you have.
6. Deliverables:
 - Copy-paste your code into the HW3 assignment:
 - Take a screenshot of each of the following test cases:
 1. *myChange* = [12, 10, 5, 19] //You have 12 Quarters, 10 dimes, 5 nickels, 19 pennies
 2. *myChange* = [8, 14, 6, 22] // You have 8 Quarters, 14 dimes, 6 nickels, 22 pennies

```
var valuesOfCoins = [
    "Quarter": 0.25,
    "Dime": 0.10,
    "Nickel": 0.05,
    "Penny": 0.01
]
```

```
var myChange: [Int] = [12,10,5,19]
var totalValue: Double = 0
```

```
if let valQuarter = valuesOfCoins["Quarter"] {
    totalValue += valQuarter * Double(myChange[0])
}
if let valDimes = valuesOfCoins["Dime"] {
    totalValue += valDimes * Double(myChange[1])
}
if let valNickel = valuesOfCoins["Nickel"] {
    totalValue += valNickel * Double(myChange[2])
}
```

```

if let valPenny = valuesOfCoins["Penny"] {
    totalValue += valPenny * Double(myChange[3])
}

```

```

print("You have a total of: $\(totalValue)")

```

A screenshot of a Swift Playground window titled "HW3" with a subtitle "No Selection". The code defines a dictionary of coin values and an array of coin counts. It then iterates through the array, calculating the total value for each coin type and printing the final result.

```

1 var valuesOfCoins = [
2     "Quarter": 0.25,
3     "Dime": 0.10,
4     "Nickel": 0.05,
5     "Penny": 0.01
6 ]
7
8 var myChange: [Int] = [12,10,5,19]
9 var totalValue: Double = 0
10
11 if let valQuarter = valuesOfCoins["Quarter"] {
12     totalValue += valQuarter * Double(myChange[0])
13 }
14 if let valDimes = valuesOfCoins["Dime"] {
15     totalValue += valDimes * Double(myChange[1])
16 }
17 if let valNickel = valuesOfCoins["Nickel"]{
18     totalValue += valNickel * Double(myChange[2])
19 }
20 if let valPenny = valuesOfCoins["Penny"] {
21     totalValue += valPenny * Double(myChange[3])
22 }
23
24 print("You have a total of: $\(totalValue)")

```

The output at the bottom of the playground is: "You have a total of: \$4.44".

A screenshot of a Swift Playground window titled "HW3" with a subtitle "No Selection". The code is identical to the previous one, but the coin counts array is different. It calculates the total value and prints the result.

```

1 var valuesOfCoins = [
2     "Quarter": 0.25,
3     "Dime": 0.10,
4     "Nickel": 0.05,
5     "Penny": 0.01
6 ]
7
8 var myChange: [Int] = [8, 14, 6, 22]
9 var totalValue: Double = 0
10
11 if let valQuarter = valuesOfCoins["Quarter"] {
12     totalValue += valQuarter * Double(myChange[0])
13 }
14 if let valDimes = valuesOfCoins["Dime"] {
15     totalValue += valDimes * Double(myChange[1])
16 }
17 if let valNickel = valuesOfCoins["Nickel"]{
18     totalValue += valNickel * Double(myChange[2])
19 }
20 if let valPenny = valuesOfCoins["Penny"] {
21     totalValue += valPenny * Double(myChange[3])
22 }
23
24 print("You have a total of: $\(totalValue)")

```

The output at the bottom of the playground is: "You have a total of: \$3.9200000000000004".

Activity #4

7. Write a simple swift program that manages the price of your cryptocurrency portfolio:

- Write a dictionary with the following key-value pairs that uses the name (string) of the coin as a key, and it's dollar value (double) as the value:
 - ByteCoin, \$60,000.00
 - CatCoin, \$0.80
 - Netherium, \$4,500.00
- Create an array called *myCoins* containing how many of each coin you have in your hypothetical portfolio.
- Write a series of functions that will use optional binding to reference the dictionary values by key and multiply them by each sequential item in your array and return the total dollar amount that you have.

8. Deliverables:

- Copy-paste your code into the HW3 assignment:
- Take a screenshot of each of the following test cases:
 1. myCoins = [0.0014, 351.1, 0.345] //you have 0.0014 ByteCoin, 351.1 CatCoin, and 0.345 Netherium
 2. myCoins = [0.0122, 241.1, 0.112] //you have 0.0122 ByteCoin, 241.1 CatCoin, and 0.112 Netherium

```
var valueOfCrypto = [  
    "ByteCoin": 60000.0,  
    "CatCoin": 0.80,  
    "Netherium": 4500.0  
]
```

```
var myCoins: [Double] = [0.0014, 351.1, 0.345]  
var totalPortfolio: Double = 0
```

```
if let valByteCoin = valueOfCrypto["ByteCoin"] {  
    totalPortfolio += valByteCoin * myCoins[0]  
}  
if let valCatCoin = valueOfCrypto["CatCoin"] {  
    totalPortfolio += valCatCoin * myCoins[1]  
}  
if let valNetherium = valueOfCrypto["Netherium"] {  
    totalPortfolio += valNetherium * myCoins[2]  
}
```

```
print("You have a total of: $\(totalPortfolio)")
```



```
1 var valueOfCrypto = [  
2     "ByteCoin": 60000.0,  
3     "CatCoin": 0.80,  
4     "Netherium": 4500.0  
5 ]  
6  
7 var myCoins: [Double] = [0.0014, 351.1, 0.345]  
8 var totalPortfolio: Double = 0  
9  
10 if let valByteCoin = valueOfCrypto["ByteCoin"] {  
11     totalPortfolio += valByteCoin * myCoins[0]  
12 }  
13 if let valCatCoin = valueOfCrypto["CatCoin"] {  
14     totalPortfolio += valCatCoin * myCoins[1]  
15 }  
16 if let valNetherium = valueOfCrypto["Netherium"] {  
17     totalPortfolio += valNetherium * myCoins[2]  
18 }  
19  
20 print("You have a total of: ${totalPortfolio}")
```



You have a total of: \$1917.3799999999999

```
1 var valueOfCrypto = [  
2     "ByteCoin": 60000.0,  
3     "CatCoin": 0.80,  
4     "Netherium": 4500.0  
5 ]  
6  
7 var myCoins: [Double] = [0.0122, 241.1, 0.112]  
8 var totalPortfolio: Double = 0  
9  
10 if let valByteCoin = valueOfCrypto["ByteCoin"] {  
11     totalPortfolio += valByteCoin * myCoins[0]  
12 }  
13 if let valCatCoin = valueOfCrypto["CatCoin"] {  
14     totalPortfolio += valCatCoin * myCoins[1]  
15 }  
16 if let valNetherium = valueOfCrypto["Netherium"] {  
17     totalPortfolio += valNetherium * myCoins[2]  
18 }  
19  
20 print("You have a total of: ${totalPortfolio}")
```



You have a total of: \$1428.88