# Learning Outcome 3 Selection Statements and Loops

Basic if statements like Java with two differences ( ) not required around boolean expresssion but can be included. The curly braces MUST be included around the then part of code.

if hScore > vScore

{

print("Home wins!")

}

OR

if ( hScore > vScore )

{

print("Home wins!")

}

Use else if or nested if statements as well.

if hScore > 4

{

println("Lots")

}

else if hScore == 0

{

println("No goals! :( ")

}

else

{

println("Something else")

}

Relational and logical operators same as Java. == != > >= < <= ! && ||

Two additional operators called identity operators: === !== Check for identical objects in memory - == checks for equal values. See separate document Example of class with == function 🡪 probably want to talk about it later with classes. Note: need to write a specific == function to be able to check for equality of object contents.

**Student Practice:**

Given two variables, money and haveHomework, write an if structure to do the following. If you haveHomework is true then display “Stay home and do homework!”, if haveHomeWork is false and money is greater than or equal to 10 and less than 20 then display “You can go to the movies!”, and if haveHomework is false and money is 20 or greater then display “You can go to the movies with a date, if haveHomeWork is false and money less than 10 then display “You have to stay home and watch TV”.

if haveHomework

{

println( "Stay home and do homework!")

}

else

{

if money < 10

{

println("You have to stay home and watch TV")

}

else if money < 20

{

println("You can go to the movies!")

}

else

{

println("You can go to the movies with a date!")

}

}

## Switch Statements

Similar to Java but break is not required in cases. Cases must be exhaustive – all cases covered so if your cases do not cover all cases you must include the default.

Can use ranges 0…40 or 0..<40. Can also use where clauses. You can specify a variable which you use in the where clause.

var grade : Int

grade = 75

switch( grade )

{

case 0..<50: print("Sorry you failed")

case 50..<60: print("Sup range")

case 60...100: print("Congratulations! You passed")

default: print("Invalid grade")

}

grade = -45

switch( grade )

{

case var g where g < 0 || g > 100: print("Invalid grade")

case 0..<50: print("Sorry you failed")

case 50..<60: print("Sup range")

case 60...100: print("Congratulations! You passed")

default: print("never get here")

}

**Student Practice:** Write a switch statement to determine the interested charged on a loan. The amount of interest charged is dependent on the loan amount. A loan amount of 100 to less than 5000 is charged 2%, a loan amount of 5000 to less than 50000 is charged 3%, 50000 to 1000000 is charged 4%, loans less than 100 are considered invalid so display invalid message, loans greater than 1000000 are consider a poor risk so display message “Loan denied - too much risk”

Sample Solutuion:

var loanAmount = 1000001

switch( loanAmount )

{

case var l where l < 100 : println("Invalid amount!")

case 100..<5000 :

let inter = Double( loanAmount ) \* 0.02

println( "Interest is \(inter)" )

case 5000..<50000 :

let inter = Double( loanAmount ) \* 0.03

println( "Interest is \(inter)" )

case 50000...1000000 :

let inter = Double( loanAmount ) \* 0.04

println( "Interest is \(inter)" )

default : println("Loan denied - too much risk")

}

## Ternary Conditional Operator

You can use ternary conditional operator in swift. Works exactly the same as in Java.

let result = numb % 2 == 0 ? "Even" : "Odd"

You can nest ternary conditional expressions too.

# Repetition

Swift has for, while and do while loops. The while and do while loops are pretty the same as Java – other than parentheses around boolean condition are optional, curly braces are ALWAYS required and no semicolon needed after while in do while loop.

var count = 0

while count < 10

{

//Note: ++ was removed in Swift 3.0

count += 1

print( count )

}

Parenthesis around boolean expression are optional.

The for loop has some differences from Java. Specify a range:

for count in 1...10

{

print( count )

}

OR

for count in 1..<10

OR 0 if no counter variable needed just use \_

var number = 2

//Note start at 2 since set number to 2 initially

for \_ in 2...10

{

number \*= 2

}

print( number )

Loop for characters in Unicode character set. Cannot do ++ on a character like in Java. Also, cannot set character based on its decimal value like in Java but you can use UnicodeScalar class.

var result : String = ""

for index in 1...6000

{

var a = Character( UnicodeScalar( index )! )

//property description is String representation of object

result += a.description + " "

if index % 30 == 0

{

print( result )

result = ""

}

}

Loop through characters in a String:

var myName = "Sharon"

for index in myName.characters.indices

{

print( myName[index] )

}

//Note: index is an object not just a integer value so you cannot do myName[3] for example

**Student Practice:** Write code to display all factors for each number from 1 to 100. Sample output:

1: 1: 1

2 : 1, 2

3: 1, 3

4: 1, 2, 4

for numb in 1...100

{

var result : String = "\(numb): "

for factor in 1...numb

{

if numb % factor == 0

{

result += "\(factor)"

if factor != numb

{

result += ", "

}

}

}

print( result )

}