(Dixit, 2023)

A person in a suit

Description automatically generated

ICL-1315 Introduction to Programming

TOBY BEEVERS

*Cambria ID: 21075206 Bangor ID: 500704883*

APPLIED DATA SCIENCE DEGREE APPRENTICESHIP BSC

Contents

[Abstract 2](#_Toc154763206)

[What are Variables? 2](#_Toc154763207)

[The use of variables in Java, Python, and C# 2](#_Toc154763208)

[Data Types 2](#_Toc154763209)

[Variable Naming 3](#_Toc154763210)

[Scope of Variables 3](#_Toc154763211)

[The use of conditional statements in Java, Python, and C# 4](#_Toc154763212)

[Syntax 4](#_Toc154763213)

[Statements 4](#_Toc154763214)

[The Growth Programming Languages 4](#_Toc154763215)

[References 5](#_Toc154763216)

[Appendix 6](#_Toc154763217)

[Program One 6](#_Toc154763218)

[Program Two 8](#_Toc154763219)

# Abstract

The use of all three programming languages, Java, C# and Python is dependent on the use case, developers are encouraged to take time to understand the differences and apply these to the use case before jumping into development.

All three languages are Object Oriented with Python also being used for Machine Learning & AI, as well as automation and scripting of repetitive tasks. C# also offers Component Oriented programming which allows a slightly different approach to a large executable binary approach.

Some would say that Java is the only 100% Object Oriented Programming Languages, however it is argued by Anuradha Dixit on Coding Ninjas that Java’s backbone is not 100% Object Oriented, sighting the use of primitive data type like Boolean, Floats and Integers outside of Objects as the primary argument.

(Dixit, 2023)

# What are Variables?

Variables can be thought of as a container for storing things that will be manipulated or referred to by a computer program. A Variable is also named to help define the information held within it, the information must be standardised, and the variable can only store a single type of information.

## The use of variables in Java, Python, and C#

To begin with Python is defined as a Dynamic language whereas Java and C# are both considered Static Languages, the main difference being that Static Languages are required to be *Compiled* where they are converted to an executable, machine-readable, low-level code. Dynamic Languages are *Interpretated* and use an interpreter to convert them to machine readable code, this has the advantage of being able to use Python across operating systems.

### Data Types

Static Languages required their Variables to be *Declared* or explicitly defined at the point of creation and cannot be changed once defined. Dynamic Languages define their data types as they are being interpreted allowing a wider versatility particularly for beginners.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Integer** | **Boolean** | **String** |
| **Java** | int age = 25; | boolean isRaining = true; | String message = "Hello, world!"; |
| **C#** | int age = 25; | bool isRaining = true; | string message = "Hello, world!"; |
| **Python** | age = 25 | is\_raining = True | message = "Hello, world!" |

Within Python you can Cast your variable values enforcing the data types that are collected within them, this provides a similar approach to Java and C#, however Python is still interpreted in the same way.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Integer** | **Boolean** | **String** |
| **Python** | age = int(25) | is\_raining = bool(True) | message = string("Hello, world!") |

### Variable Naming

Naming conventions for variables is also an important consideration when looking across languages, whilst standard approaches of camelCase or PascalCase or snake\_case can be used, the different languages do have different requirements, some demand certain conventions others are less prescriptive but do have a best practice.

|  |  |  |
| --- | --- | --- |
|  | **Variables** | **Constraints** |
| **Java** | Uses camelCase  int myVariable; | Uses UPPERCASE  final int MAX\_VALUE = 100; |
| **C#** | Uses PascalCase  int MyVariable; | Uses PascalCase  const int MaxValue = 100; |
| **Python** | Uses snake\_case  my\_variable = 10 | Uses UPPERCASE  MAX\_VALUE = 100 |

All three languages follow the same principal restrictions for naming variables.

* Variable names cannot start with a digit like 9 or 3.
* Special characters other than the underscore are not permitted, example # % & etc.
* Keyword words like if, class, while, etc cannot be used as variable names as these are reserved.

(Oracle, 2023) (Python.Org, 2023) (Microsoft, 2023)

### Scope of Variables

Variable scopes determine where in a program a variable is accessible and usable, Java and C# use similar approaches whilst Python is totally different in its approach.

##### Java & C#

* **Local Variables:** Defined and accessible only within the declared scope.
* **Instance Variables:** Belong to an instance of a class and accessible throughout the class but unique to each instance.
* **Static Variables:** Belong to the class itself and shared among all instances of the class.

##### Python

* **Local Variables:** Defined within a function or method and accessible only within the function they are defined in.
* **Global Variables:** Defined outside functions or in the global scope and accessible throughout the module.

It could be argued that Static Variables in both C# and Java and Global Variables in Python are similar as they are both defined outside of the scope of the class itself and can be shared.

(Visual Studio Code, 2023)

# The use of conditional statements in Java, Python, and C#

Conditional Statements are a key part of writing a program. As the program follows the algorithm, conditional statements are used to make decisions for example if a value meets a certain criteria then the answer is yes.

*Example*

a = 33  
b = 200  
if b > a:  
  print("b is greater than a")

The main difference between the three languages is their syntax, in terms of what they deliver in functionality it is exactly the same. It’s worth noting that the syntax differences do make them unique in their implementation and usage.

## Syntax

* Java uses curly braces {} to denote code blocks within conditional statements.
* C# also uses curly braces {} like Java for denoting code blocks within conditional statements.
* Python uses indentation to denote code blocks within conditional statements.

Unlike Java and C#, in Python indents are crucial for the structure and readability of the code.

Java and C# also require a semi colon to end each statement, this helps to prevent syntax ambiguities. Whereas Python simply allows a new line to end the statement.

## Statements

Both Java and C# allow for the use of a switch statement, this is lacking in Python often leading to long chains of if/elif to handle multiple conditions.

The switch statement in Java and C# allows the developer to define a variable containing the expression just once, the program than uses a case statement to validate the expression. Once the expression is validated as a match the block of code associated with the case statement will be executed.

In my humble opinion Python seems to be much easier language for beginners, the use of indenting increases Pythons readability allowing the reader to better understand the flow of the code.

# The Growth Programming Languages

In your own words which programming language out of the three (C#, Java, and Python) do you feel is growing and has the best employment opportunities and why?

# References

Dixit, A., 2023. *Is Java an Object-Oriented language?.* [Online]   
Available at: https://www.codingninjas.com/studio/library/is-java-an-object-oriented-language  
[Accessed 28 12 2023].

Microsoft, 2023. *.NET documentation.* [Online]   
Available at: https://learn.microsoft.com/en-us/dotnet/  
[Accessed 28 12 2023].

Oracle, 2023. *Variables.* [Online]   
Available at: https://docs.oracle.com/javase/tutorial/java/nutsandbolts/variables.html#:~:text=A%20variable's%20name%20can%20be,%24%20%22%20or%20%22%20\_%20%22.  
[Accessed 28 12 2023].

Python.Org, 2023. *Type Variable Names.* [Online]   
Available at: https://peps.python.org/pep-0008/#naming-conventions  
[Accessed 28 12 2023].

Visual Studio Code, 2023. *Programming Languages.* [Online]   
Available at: https://code.visualstudio.com/docs/languages/overview  
[Accessed 28 12 2023].

# Appendix

## Program One

A program that includes a selection of a variable type from a user, the use of IF statements and include at least one nested IF that will output the definition and usage of the selected variable type. **You are to research the variable definitions and paraphrase them into your own wording.**

#Create a print statement to direct the user to select from the list of choices 1, 2, 3, or 4 etc

print ("Select a Variable Type from this list to get started:")

print () #added an empty print statement to add a space

print ("1 to understand more about Numeric Types,")

print ("2 to understand Text,")

print ("3 to understand Sequence Types,")

print ("4 to understand Mapping Types,")

print ("5 to understand Set Types ")

print ("6 to understand Boolean Types")

print ("7 to understand List & Tuple Types")

print () #added an empty print statement to add a space

#Create a variable to store the user's choice, force and integer value

userChoice = int(input("Enter Your Selection between 1-7: "))

print () #added an empty print statement to add a space

#Add the if & elif, print statement to display the user's choice and add else to capture incorrect User Selection

if userChoice == 1: #Numbers

print ("You have selected to know more about Numbers in Python!")

print () #added an empty print statement to add a space

print ("Numbers can be Integers, Floating Point Numbers, or Complex Numbers")

print () #added an empty print statement to add a space

print (" - Int, or integer, is a whole number, positive or negative, without decimal.")

print (" - Float, or \"floating point number\" is a number, positive or negative,")

print (" containing one or more decimals.")

print (" - Complex is an advanced number, which take a real number and an imaginary number.")

elif userChoice == 2: #Text

print ("You have selected to know more about Strings also know as Text in Python!")

print() #added an empty print statement to add a space

print ("Strings in python are surrounded by either single quotation marks, or double quotation marks.")

print ("'Hello' is the same as \"hello\" You can display a string literally with the print() function")

print ("A str can be letters of numbers but it's important to understand that numbers")

print ("stored in str can not be calculated!")

elif userChoice == 3: #Sequence Types

print ("You have selected to know more about Sequence Types in Python!")

print () #added an empty print statement to add a space

print ("An advanced data type, the sequence data type in Python is used to store")

print ("sequential data. The sequence data type in python can be considered")

print ("as a container that can store different data.")

elif userChoice == 4: #Mapping Types

print ("You have selected to know more about Mapping Type in Python!")

print () #added an empty print statement to add a space

print ("An advanced data type, a mapping type is a data type comprised of a collection")

print ("of keys and associated values. Python's only built-in mapping type is the")

print ("dictionary. Dictionaries implement the associative array abstract data type.")

elif userChoice == 5: #Set Types Lists & Tuples

print ("You have selected to know more about Set Types in Python!")

print () #added an empty print statement to add a space

print ("Sets, Tuples & Lists are all similar in that they are ordered collections of data")

print (" - A Set is a collection of unique values stored in a single variable.")

print (" A Set is a collection which is unordered, unchangeable, and unindexed.")

print () #added an empty print statement to add a space

print (" - A Tuple is a collection of values that are ordered.")

print (" A Tuple is a collection which is ordered, unchangeable, and indexed.")

print () #added an empty print statement to add a space

print (" - A List is a collection of values that are ordered and change.")

print (" A List is a collection which is ordered, changeable, and indexed.")

elif userChoice == 6: #Boolean Types

print ("You have selected to know more about Boolean Type in Python!")

print () #added an empty print statement to add a space

print ("In programming you often need to know if an expression is True or False,")

print ("You can evaluate any expression in Python, and get either, True or False.")

print ("When you compare two values, the expression is evaluated and")

print ("Python returns the Boolean answer")

else:

print ("You have selected an invalid choice, your selection must be between 1-6")

exit()

# I have exited the program here, I know I could add a Loop to get the user back to the start however that wasn't requested in the assignment

## Program Two

A program that will carry out several forms of calculations based on user inputs and include IF statements that incorporate the ‘AND’ (or ‘OR’) operator.

#import the DataTime Library and set a variable with todays date

import datetime

dateNow = datetime.datetime.now()

# Create a Program that will capture a Users Age and return information about Key Events

# Ask the User to enter their DOB

print ("Would you like to know some interesting facts about your age?")

print () # Print a blank line

userAge = str(input("Enter your Date of Birth DD-MM-YYYY: ")) # Capture the Users DOB in this variable

# This variable stores the date after is has been converted into a usable date format using the datatime library > strptime function

# https://docs.python.org/3/library/datetime.html?highlight=datetime#module-datetime

parsedDate = datetime.datetime.strptime(userAge, "%d-%m-%Y").date()

#print (parsedDate) #Keep as debug test to see if the date is parsed correctly

#now start returning values

print () # Print a blank line

print ("So you are: ")

#Variable to store the age of the user in years

userYearsOld = dateNow.year - parsedDate.year # Calculate the Users Age in Years

print (" \* " +str(userYearsOld) + " Years Old")

#Varaible to store the age of the user in days

numberOfDaysOld = (dateNow.date() - parsedDate).days # Calculate the Users Age in Days

userDaysOld = numberOfDaysOld - dateNow.day # Calculate the Users Age in Days

print (" \* That's " + str(userDaysOld) + " Days Old")

#Add some calc to use in if & and statements

if userYearsOld >= 18:

print () # Print a blank line

print ("And")

print (" \* It looks like your old enough to vote!")

print (" \* You can grab a beer at the bar!")

print (" \* Plus you can also rent a car!")

elif userYearsOld < 18:

userAgeCalc = 18 - userYearsOld

print () # Print a blank line

print ("Not quite there, only " + str(userAgeCalc) + " years until you can vote & grab a beer!")

#Add some trivia statements

print () # Print a blank line

print ("Did you know:")

print (" \* Your age in dog years is over " + str(userYearsOld \* 7))

print (" \* Your age in cat years is over " + str(userYearsOld \* 4))

print (" \* You probably had somewhere near " + str(round(userDaysOld / 7)) + " Sunday Lunches!")

#Add calcs about Driving ages if & and statements

print () # Print a blank line

print () # Print a blank line

print ("What about Driving:")

if userYearsOld < 16:

print (" \* Sorry you can't start Driving just yet!")

elif userYearsOld >= 16 and userYearsOld <= 70:

print (" \* Yep, you're good with driving as long as you've past your test!")

elif userYearsOld > 70:

print (" \* Did you know you need to be retesting every three years to keep you Licence?")

exit ()