CPSC Midterm 1 preparation

# GIT version control

Version Control System (VCS) keeps tracks of changes made to a set of files over time.

## 3 types of VC

* Local VCS
  + Manages changes of files in local computer.
* Centralized VCS
  + Versions of files in a single server. Client must check out files from server. Changes are made on the server.
* Distributed VCS
  + Clients may have copies of the versions of files and are able to update versions on server. Git.

## Git Terminology

* Working tree: what is seen on the file system
* Staging area: stores the files that are tracked and ready to be committed. The **git add** command moves the files from working tree to the staging area.
* History: stores all changes made in the *.git* hidden directory. After each **git commit**, the history is updated.

## Tracking a local project

* ***Ls -la***: to list the file’s contents.
* ***Git init***: to initialize empty Git repository.
* ***Rm -rf .git***: to stop tracking a project.
* ***Git status***: to check status of a project
* ***Git add <<doc***.***type>>***: to track a file in project
* ***Git add –all***: to track all files in project
* ***Git commit -m ‘txtxtxtxt’:*** to make a commit with a comment
* ***Git log***: to check all versions of the tracked file
* ***Git reset <<doc***.***type>>***: to remove a file from a staging area.
* ***Git reset***: to remove everything from the staging area.
* ***Git checkout <<previous-hash-value>>:*** to revert previous version
* ***Git pull:*** to update your local project with the server’s information.
* ***Git push:*** to feed the server with the work in the local computer.

# Python basics

Python is a simple and practical Object-Oriented Programming Language. Good for Data Science, Security, Web Programming, Databases.

## Read Evaluate Print Loop (REPL)

* Interactive shell (Command Line Interface)
* Jupyer notebook (GUI)

## Integrated Development Environment (IDE)

* Text editor
* Debugger
* Autmation tools
* IntelliSense

## Keywords

**Table

Description automatically generated**

## Identifiers

An identifier is a programmer-defined name consisting of letters a to z and A to Z, digits 0, 1 to 9, underscore \_.

* CANNOT be a Python keyword
* CANNOT start with a digit
* CAN have any length

## Naming Conventions

Table

Description automatically generated

## Data Types

Graphical user interface, application

Description automatically generated

## Python Comments

* **In-line comments**: go after a code line
* **Block comments**: 2 or more lines before the code that provide some explanation
* **Documentation string comments**: Go before the whole program. They are enclosed with three single quotes before and after.

## Binary operators

Table

Description automatically generated

Table

Description automatically generated

Table

Description automatically generated

# Data structures

Diagram

Description automatically generated

## Lists

* Data structure that stores a collection of data as an ordered sequence in a continuous memory location.
* A list can hold different data types.
* Mutable.
* Commands
  + **len()**: to know the length of the list.
  + **append()**: to add an element to the list.
  + **pop()**: to remove the last element from the list.
  + **del list\_name[i]**: remove an element at position number i.
  + **list\_1 + List\_2**: to add two different lists.
  + **x in list\_name**: to check if x is an element in list.
  + **list**.**sort()**: to sort a list in ascending order.
  + **list**.**sort(reverse = True)**: to sort a list in descending order.
* List slicing:
  + Text

    Description automatically generated

## Tuple

* Data structure that stores a collection of data as an ordered sequence in a continuous memory location.
* A list can hold different data types.
* Immutable.
* Elements can be accessed with positive or negative indices, like a list.

## Dictionary

* A dictionary is a key Python data structure that stores a collection of key-value pairs (KVPs)
* Elements of the dictionary values are accessed using keys

Diagram, timeline

Description automatically generated

# Control Structures

Text

Description automatically generated

## Selection / Decision / Conditional

### If tests

Text

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with medium confidence

Graphical user interface, text

Description automatically generated

### Match case statement

Text

Description automatically generated

## Iteration / Repetition / Loop

### For Loop

For loop iterates over a sequence and mapping data types (string, list, tuple, dict)

A screenshot of a computer

Description automatically generated with medium confidence

### While Loop

While loop is used to iterate over a block statement as long as the *while* condition is true.

Infinite while loop must be avoided by

* ensuring the while condition will be become False
* exiting the loop after specific number of iterations

Text

Description automatically generated

### Range method

The Python range() method has the following syntax:

**range(start, end, step)**

* Returns a sequence starting from start. Default is zero.
* Increments by step. Default is one.
* Stops at end-1. No default.

Text

Description automatically generated

### List comprehension (For Loop + Range)

Text

Description automatically generated

## Random module

Text

Description automatically generated

# Midterm 1 questions

## Generate 50 random numbers from 100 to 200.

## Generate 20 three-digit numbers that are divisible by 50.

## Create a tuple with the first 20 powers of 2 using a For Loop. (2\*\*0=1, 2\*\*1=2, 2\*\*2=4 …. 2\*\*19=524,288