**CCP SESSION**

1. Parts of Programming

* Sequence
  + Variables – identifier given to memory where data will be stored
  + Input
  + Output
  + Operators
* Branching
* Loop

1. Basic Castings in Python
   1. int
   2. float
   3. bool
   4. str
   5. complex
2. Python Character Set: UNICODE (Universal Code) is a 16 bit-code that supports 24 types of languages.
3. OOPS: INPUT 🡨 PROCESS 🡨 OUTPUT
4. Updating data in python results in new memory location of the data.
5. AGC – Auto Garbage Collection (unused and non – referring data gets deleted)

# Program to get false on input

flag = bool(input(“data”))

print(“flag is”, flag)

* Branching

|  |  |
| --- | --- |
| **if** | **Within loop** |
| if-else | End of code |
| if-elif-else | Multiple |

1. If it is perfectly divisible 🡪 use mod operation
2. In multiple ‘and’ condition, it will stop on first false

If\_\_\_and\_\_\_and\_\_\_:

1. In multiple ‘or’ condition, it will enter on first True

If\_\_or\_\_or\_\_:

1. Nested if and logical if both are same in complexity. But in terms of readability logical if is preferred
2. Python doesn’t support i++/ ++i It supports i = i+1 , i+=1
3. Loop
   1. Start (initial value)
   2. Condition (where to stop)
   3. Step (+k or -k mostly)
4. Difference between while and for is:

|  |  |
| --- | --- |
| While | Loop |
| While can have fractional steps | For can only have integer step |

1. QR Code (Quick Response Code)
2. Primitive value 🡪 passed by value
3. Data Structure 🡪 reference

**Python: OOPS**

* **Encapsulation:** process of combining (data + operation) in a unit
* **Abstraction:** process of giving access to data in minimum less complicated manner (give only access hide the way it is done, as user is not concern with it)
* **Inheritance:** process of taking properties from pre-existing and creating something new (own properties = inherited properties)
  + Positive
  + Faster Development
  + Enhancement
  + Evolution
  + Standardization
* **Polymorphism:** ability to exist in multiple forms
  + Overload 🡪 1 function name with n operations(data)
  + Override 🡪 sub class rewrites super class’s methods

|  |  |
| --- | --- |
| **Class** | **Object** |
| Virtual Idea | Physical Implementation |
| Has properties | Has values |
| Operations on properties  (Setter, getter, display) |  |
| Gives framework | Implementation (code) |
| One class of N objects | N objects of One class |

* **Constructor:** is a function that is called on its own when object is created.
  + Syntax 🡪 def \_\_init\_\_(self):
  + Used for: Initialization, Default Value, Validation
* \_\_XXXXXX\_\_ 🡪 **System Methods**
* **Del method:** called on its own just before anything goes for garbage collection
  + **Syntax** 🡪 def \_\_del\_\_(self):
* **Access Specifiers/ Visibilities:**
  + **Public 🡪** var(default)
  + **Private** 🡪 \_var(not visible but known can access)
  + **Strong Private 🡪** \_\_var(not visible not accessed outside class)
* Strong Private is not inherited
* Is a: inheritance has a: containership

**Exception Handling:**

**Graphical user interface, text, application, email

Description automatically generated**

* **Algorithm**

Finite set of instructions if followed will solve given problem in finite time in finite steps