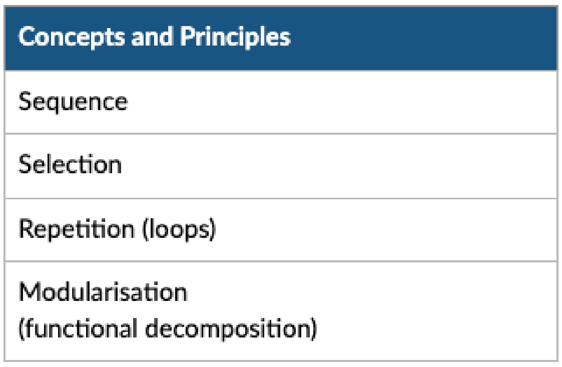
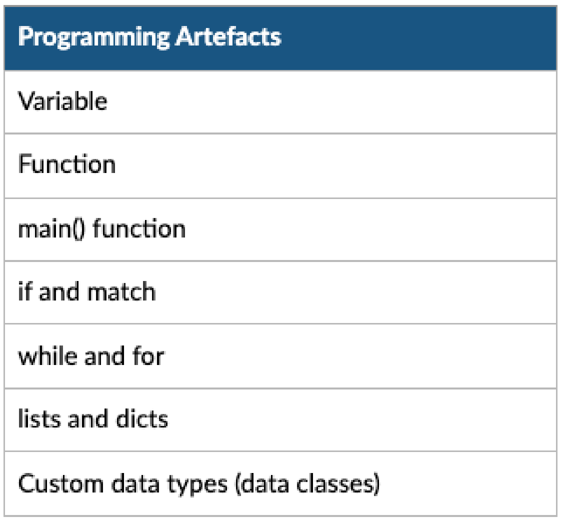


Concepts and Principles
Sequence
Selection
Repetition (loops)
Modularisation (functional decomposition)



Programming Artefacts
Variable
Function
main() function
if and match
while and for
lists and dicts
Custom data types (data classes)

Sequence

A sequence in programming is a way in which it will run or execute, be it from top to bottom, left to right, aesc or desc.

Example:

```
odd_nums = [1,3,5,7,9]
print(“Hello”)
print(“Everyone”)
print(“!”)
print(f”{odd_nums}”)
```

Selection

A selection is programming is a decision of which line of code should run under a given condition.

Example:

```
if 10%2 == 0:
        print (“given number is even”)
else
        print (“given number is odd”)

num = 10
match num:
        case “5”
                print (“Better luck next time!”)
        case “10”
                print (“found 10!”)
```

Repetition (loops)

Repetition is a technique of continuous execution of a line or a block of code until the range or condition is changed.

Example:

```
for i in range (1,10)
        print (i)

num = 1
while num>10:
        print(“True”)
        num += 1
```

Modularisation(functional decomposition)

Modularisation in programming is a structure that is followed to break complex programs into meaningful functions which improve code readability, scalability and maintenance.

Example:

```
def add_nums(int a, int b) -> int
        return a+b

add_nums(6,5)
```

Variables

Variables in programming are like containers which store the reference for the data stored in the memory.

Example:

```
name: str = “Sachin”
```

This *name* variable will store the reference for the string Sachin stored in the memory.

Function

Functions in programming are reusable blocks of code that can be defined and used multiple times in different parts of the program. It helps to break complex concepts into meaningful blocks which improve code readability, scalability and maintenance.

Example:

```
def add_nums(int a, int b) -> int
        return a+b

add_nums(6,5)
```

main() function

main() function is literally the “main” function of in python as the name suggests, which runs when the program is executed, unlike other functions it is called differently, like:

```
if __name__ == "__main__":
        main()
```

if and match

if in programming comes under selection where it matches the code conditionally, when one or the other condition is matched. It is mainly used where there are minimal cases.

Example:

```
if 10%2 == 0:
        print (“given number is even”)
else
        print (“given number is odd”)
```

match in programming also comes under selection where it matches the code as a pattern, when one or the other pattern is matched. It is mainly used where there are several cases.

Example:

```
num = 10
match num:
        case “5”
                print (“Better luck next time!”)
        case “10”
                print (“found 10!”)
```

while and for

while and for in python comes under repetition where they are used to continuously execute a line or a block of code until the range or condition is changed.

Example:

```
for i in range (1,10)
        print (i)

num = 1
while num>10:
        print("True")
        num += 1
```

lists and dicts

lists in python are ordered collections which are used to store data in it where its values are indexed by the indexes staring at 0.

Example:

```
odd_nums = [1,3,5,7,9]
```

dicts in python are key-value type storage where the key acts as indexes that are connected with its value. Keys can be initialised by the user with their values.

Example:

```
my_details = {“name”: “Sachin”, “ID”: 689206}
```

Custom data types (data classes)

Custom data types (data classes) are used when we need to capsulise related data in one object, which then can be used in multiple parts of the program just by calling the data class with its related fields.

Example:

```
@dataclass
class University:
  name: str
  est: int
  rank: int

my_car = Car("UTAS", 1980, 1)
```