IF-ELIF-ELSE

If-Elif-Else is a class of statements that are known as conditional statements.

They accept a conditional expression (which we can also call as only ‘condition’ or only ‘expression’), and process a piece of code based on that.

A conditional statement evaluates to True or False.
Note: In Python, 0 is considered as False and 1 is considered as True.

IF-ELSE Statement in Code

Simple if-else program to check voter eligibility:

```
age = int(input("Enter your age: "))

if age >= 18:
    print("Eligible to vote")
else:
    print("Not eligible to vote")

# One more example
a = 9
b = 2
if a > b :
    print("a is greater than b")
```

ELIF Statement in Code (Part 1)

```
# Elif stands for "else if"

a = 33
b = 32
if b > a:
  print("b is greater than a")
elif a == b:
  print("a and b are equal")
else:
  print(" b is less than a")
```

ELIF Statement in Code (Part 2)

```
day = input("Enter a week day's name: ")

if day == "Sunday":
    print(1)
elif day == "Monday":
    print(2)
elif day == "Tuesday":
    print(3)
elif day == "Wednesday":
    print(4)
elif day == "Thursday":
    print(5)
elif day == "Friday":
    print(6)
elif day == "Saturday":
    print(7)
else:
    print("Enter a valid day like Sunday, Monday,...")
```

Problem

```
var = 0
if var:
    print("In If")
else:
    print("In Else")

var = 1
if var:
    print("In If")
else:
    print("In Else")
```

Shorthand If-Else

```
# Shorthand if else statment
# This technique is known as Ternary Operators, or Conditional Expressions

a = 2
b = 330
print("A") if a > b else print("B")
```

Here four lines of code have been reduced to one.

For Loop

Purpose of For loop is pure iteration.

For example: if we want to print first 10 natural numbers then we would simply iterate over those numbers using the range() built-in.

Code:
#1

```
for i in range(1, 11):
    print(i)
```

#2

```
l = ['Ashish', 'Lijiya', 'Bala']
for i in l:
    print(i)
```

Ashish
Lijiya
Bala

Q2: Python program to print all the even numbers within the given range.

```
x = int(input("input range: "))
for i in range(1, x):
    if i % 2 == 0:
        print(i)
```

Sum() and Average() : Way 1

Q3: Write a program to read 10 numbers from the keyboard and find their sum and average.

#Write a program to read 10 numbers from the keyboard and find their sum and average.

```
l = []
for i in range(10):
    l.append(int(input("Enter the number: ")))

s = sum(l)
print(s)

avg = sum(l) / len(l)
print(avg)
```

Sum() and Average() : Way 2 (Using statistics module)

Q3: Write a program to read 10 numbers from the keyboard and find their sum and average.

```
import statistics
#Write a program to read 10 numbers from the keyboard and find their sum and average.

l = []
for i in range(10):
    l.append(int(input("Enter the number: ")))

s = sum(l)
print(s)
m = statistics.mean(l)
print(m)
```

While Loop

While loop is an entry controlled loop.

First thing we need is a variable to control our while loop.

Second thing a while loop requires is a condition (or conditional expression).

```
if True:
    pass
```

And a condition requires: a variable and some test on it that would result in True or False.

Generally:
We let our variable to start with something like: i = 1
We let our condition be something like: i < 10

Printing first ten natural numbers

For 10 natural numbers
Let: i = 1
Condition: i <= 10

Third thing is modification of variable used in the condition above.
i += 1

Three steps described here are:
1. initialization
2. condition
3. change

Infinite Loop

You can create infinite loop by manipulating these three steps of a while loop:

1. initialization
2. condition
3. change

```
while (True):
    # Do something infinite number of times
```

Break: breaks out of the normal flow of the code

Continue: it skips the current iteration on encountering this keyword and continues with the next iteration.