Example:

```
cost=float(input("Bike Price "))
if cost>100000:
    tax=cost*15/100
elif cost>50000 and cost<=100000:
    tax=cost*10/100
else:
    tax=cost*5/100

print(f'Tax to paid is {tax:.2f}')
```

Example:

```
p=int(input("Enter precentage"))
if p>90:
    print("A")
elif p>80 and p<=90:
    print("B")
elif p>=60 and p<=80:
    print("C")
else:
    print("D")
```

Example:

```
ord('a')
97
ord('z')
122
>>> ord('A')
65
>>> ord('Z')
90
>>> chr(65)
'A'
>>> chr(66)
'B'
>>> chr(97)
'a'
```

Example:

```
# write a program to input character and convert
# character into uppercase or lowercase


ch=input("Enter any character ")
if ch>='a' and ch<='z':
    ch=chr(ord(ch)-32)
    print(f'Coversion character {ch}')
elif ch>='A' and ch<='Z':
    ch=chr(ord(ch)+32)
    print(f'Conversion character {ch}')
else:
    print(f'{ch} is not alphabet')


# ord() : returns ascii value of input character
# chr() : returns character value of input ascii value
```

Nested if

If followed by if is called nested if (OR) if within if is called nested if.

Syntax:

```
If <condition1>:
  If <condition2>:
     Statement-1
     Statement-2
  else:
     Statement-3
     Statement-4
else:
   statement-5
   statement-6
if condition1,condition2 are True, PVM executes statement-1,statement-2
if condition1 is True and condition2 False, PVM executes statement-
3,statement-4
if codnition1 is False, PVM executes statement-5,Statement-6
```

Example:

```
# Login

user=input("UserName ")
pwd=input("Password ")
if user=="nit":
    if pwd=="nit123":
        print("Welcome")
    else:
        print("invalid password")
else:
    print("invalid username")
```

Output:

```
UserName nit
Password nit123
Welcome

UserName nit
Password abc
invalid password

UserName abc
Password nit
invalid username
```

Example:

```
# Banking Logic

accno=int(input("AccountNo "))
bal=float(input("Balance "))
ttype=input("Transaction Type (D/W) ")
tamt=float(input("Transaction Amount "))
if ttype=='D':
    bal=bal+tamt
elif ttype=='W':
    if tamt<bal:
        bal=bal-tamt
    else:
        print("Insuff Balance")
else:
```

```
    print("Invalid Transaction Type")

print(f"'Account {accno}
Balance {bal:.2f}"')
```

Output:

```
AccountNo 1
Balance 50000
Transaction Type (D/W) D
Transaction Amount 5000
Account 1
Balance 55000.00

AccountNo 2
Balance 45000
Transaction Type (D/W) W
Transaction Amount 10000
Account 2
Balance 35000.00

AccountNo 3
Balance 50000
Transaction Type (D/W) W
Transaction Amount 90000
Insuff Balance
Account 3
Balance 50000.00

AccountNo 4
Balance 10000
Transaction Type (D/W) X
Transaction Amount 1000
Invalid Transaction Type
Account 4
Balance 10000.00
```

match statement

The match-case statement, also known as pattern matching, is a feature introduced in Python 3.10. It provides a concise and expressive way to

perform pattern matching on data structures, such as tuples, lists, and classes.

Syntax:

```
match(value/variable/expression):
        case <pattern>:
                statement-1
                statement-2
        case <pattern>:
                statement-3
                statement-4
        case <pattern>:
                statement-5
                statement-6
        case _:
                statement-7
                statement-8
```

_ represents default pattern, this case is executed when if input value does not match any pattern or case.

Example:

```
value=5
match(value):
    case 'V':
        print("V")
    case 5:
        print("5")
    case _:
        print("does not match")
```

Output:

```
does not match
```