Part 1

1. Evaluate the following expressions for `num1 = 10` and `num2 = 20`.

(a) `not (num1 < 1) and num2 < 10`

```
num1 = 10
num2 = 20
print(not(num1 < 1)) and (num2 <10)
```

```
C:\Users\bajra\PycharmProjects\pythonProject\v
True

Process finished with exit code 0
```

(b) `not (num1 < 1) and num2 < 10 or num1 + num3 < 100`

```
n.py
num1 = 10
num2 = 20
print(not(num1 < 1) and (num2 <10) or (num1 + num2)< 100)
```

```
main
C:\Users\bajra\PycharmProjects\pyth
True

Process finished with exit code 0
```

(c) `not (num2 > 1) or num1 > num2 - 10`

```
orkshop.py > ...
num1 = 10
num2 = 20
print (not (num2 > 1) or num1 > num2 - 10 )
```

```
PS C:\Users\
False
```

2. Give an appropriate if statement for each of the following
(The value of `num` is not important):

(a) Displays `'within range'` if `num` is between 0 and 100, inclusive.

```
num = 10
if (0<= num <=100):
    print("within range")
```

```
C:\Users\bajra\PycharmProjects\pythonProje
within range

Process finished with exit code 0
```

(b) Displays `'within range'` if `num` is between 0 and 100, inclusive, and displays `'out of range'` otherwise.

```
num = 10
if (0<= num <=100):
    print("within range")
else:
    print("out of range")
```

3. Rewrite the following if-else statements using a single if statement and elif:

```
if temperature >= 85 and humidity > 60:
    print ('muggy day today')
else:
    if temperature >= 85:
        print ('warm, but not muggy today')
    else:
        if temperature >= 65:
            print ('pleasant today')
        else:
            if temperature <= 45:
                print ('cold today')
            else:
                print ('cool today')
```

```python
temperature = float(input("Enter current temperature : "))
humidity = float(input("Enter humidity : "))
if temperature >= 85 and humidity > 60:
    print ('muggy day today')
elif(temperature >= 85):
    print ('warm, but not muggy today')
elif(temperature >= 65):
    print ('pleasant today')
elif(temperature <= 45):
    print ('cold today')
else:
    print ('cool today')
```

4. Write a Python program in which:

(a) The user enters either `'A', 'B', or 'C'`. If `'A'` is entered, the program should display the word `'Apple'`; if `'B'` is entered, it displays `'Banana'`; and if 'C' is entered, it displays `'Coconut'`. Use nested if statements for this.

```python
user_input = input("enter A,B, or C:")
if user_input =="A":
    print("Apple")
else :
    if user_input =="B":
        print("Banana")
    else:
        if user_input=="C":
            print("Coconut")
        else:
            print("invalid")
```

else › else › if user_input=="C"

main

```
C:\Users\bajra\PycharmProjects\pythonProject\venv\Scripts\pytho
enter A,B, or C:C
Coconut

Process finished with exit code 0
```

(b) Repeat question **(a)** using an if statement with `elif` headers instead.

```python
user_input = input("enter A,B, or C:")
if user_input =="A":
    print("Apple")
elif user_input =="B":
        print("Banana")
elif user_input=="C":
            print("Coconut")
else:
            print("invalid")
```

(c) A student enters the number of college credits earned. If the number of credits is greater than or equal to 90, 'Senior Status' is displayed; if greater than or equal to 60, 'Junior Status' is displayed; if greater than or equal to 30, 'Sophomore Status' is displayed; else, 'Freshman Status' is displayed.

```python
credits = int(input("enter the no of coolege credit earned:"))
if credits > 90:
    print("senior status")
elif credits >= 60:
    print("junior status")
elif credits >= 30:
    print("sophomore status")
else:
    print("freshman status")
```

else

main

```
C:\Users\bajra\PycharmProjects\pythonProject\venv\Scripts\python.exe C
enter the no of coolege credit earned: 40
sophomore status

Process finished with exit code 0
```

(d) The user enters a number. If the number is divisible by 3, the word 'Fizz' should be displayed; if the number is divisible by 5 the word 'Buzz' should be displayed and if the number is divisible by both 'FizzBuzz' should be displayed.

```python
number = int(input("enter a number :"))
if number % 3 == 0 and number % 5 == 0:
    print("fizzbuzz")
elif number % 3 ==0:
    print("fizz")
elif number % 5 == 0:
    print("buzz")
else:
    print("the no. is not divisible by 3 or 5")
```

else

main

```
C:\Users\bajra\PycharmProjects\pythonProject\venv\Scripts\pyt
enter a number : 10
buzz

Process finished with exit code 0
```

5. Sam wants to store his series of car to a list. The list of a car are: (up to you). After creating a list he add some car and delete some car and at last there are still 5 cars left in his list. Additionally, he wants his car to be shuffled every time when the list is being displayed. [Hint: shuffle from random]

```python
from random import shuffle
cars = ["mercedes benz","ferrari","tesla model y","ford mustang","range rover"]
cars.append("lamborgini hurricane")
cars.remove("ferrari")
shuffle(cars)
print(cars)
```

```
C:\Users\bajra\PycharmProjects\pythonProject\venv\Scripts\python.exe C:\Users\bajra\PycharmProjects\pyt
['range rover', 'ford mustang', 'mercedes benz', 'tesla model y', 'lamborgini hurricane']

Process finished with exit code 0
```

Part 2

1. Write a program that:

(a) Uses a loop to add up all the even numbers between 100 and 200, inclusive.

```python
even_sum = 0
i = 100
while i <=200:
    if i % 2 ==0:
        even_sum +=i

    i += 1
print(even_sum)
```

while i <=200

main

```
C:\Users\bajra\PycharmProjects\pythonProject\venv\Scr
7650
```

(b) Sums a series of (positive) integers entered by the user, excluding all numbers that are greater than 100.

```python
sum = 0
while True:
    userinput = int(input("enter a positive integer else enter 0 to end the program :"))
    if userinput ==0:
        break
    elif userinput <=100:
        sum += userinput
    else:
        print("invalid ")
print("sum:", sum)
```

```
main
C:\Users\bajra\PycharmProjects\pythonProject\venv\Scripts\python.exe C:\Users\bajra\PycharmProjects\pythonProject
enter a positive integer else enter 0 to end the program : 12
enter a positive integer else enter 0 to end the program :11
enter a positive integer else enter 0 to end the program :121
invalid
enter a positive integer else enter 0 to end the program :
```

(c) Solves Q2 but this time using an infinite loop, break and continue statements.

```python
sum = 0
while True:
    userinput = int(input("enter a positive integer else enter 0 to end th
    if userinput ==0:
        break
    elif userinput <=100:
        sum += userinput
    else:
        print("invalid ")
        continue
print("sum:",sum)
```

(e) Prompts the user to enter any number of positive and negative integer values, then displays the number of each type that were entered.

```python
positive_num = 0
negative_num = 0
while True :
    user_input = int(input("enter an integer value else 0 to end"))
    if user_input ==0:
        break
    elif user_input > 0:
        positive_num +=1
    else :
        negative_num +=1
print("total num of postive number",positive_num)
print("total number of negative number:",negative_num)
```

2. The following while loop is meant to multiply a series of integers input by the user, until a sentinel value of 0 is entered. Indicate any errors in the code given. See if you can fix the program and get it running.

```
product = 1
num = input('Enter first number: ')
while num != 0:
    num = input('Enter first number: )
    product = product * num
  print('product = ', product)
```

```
product = 1
while True:
    num = int(input("enter first num:"))
    if num!= 0:
        product = product * num
    else:
        break
print("product = ", product)
```

3. For each of the following, indicate which the definite loop is, and which an indefinite loop, explain your reasoning.

(a)

```
num = input('Enter a non-zero value:')
while num == 0:
    num = input('Enter a non-zero value: ')
```

```python
num = input("enter a non zero value:")
while num == 0:
    num = input("enter a non zero value:")
```

while num == 0

main

```
C:\Users\bajra\PycharmProjects\pythonProject\venv\Scripts\python.exe
enter a non zero value: 3

Process finished with exit code 0
```

It is indefinite loop because the number of times it is going to execute is not known.

(b)

```python
num = 0
while n < 10:
    print 2 ** n
    n = n + 1
```

```python
n = 0
while n < 10:
    print(2**n)
    n = n+1
```

while n < 10

main

```
8
16
32
64
128
256
512

Process finished with exit code 0
```

It is definite loop because the time when program will execute is known.

Part 3

1. Create three dictionaries:

```
dic1 = {1:10, 2:20}
dic2 = {3:30, 4:40}
dic3 = {5:50, 6:60}
```

(a) Write code to concatenate these dictionaries to create a new one. Create a variable called `nums` to store the resulting dictionary. There are multiple ways to do this, however, one of the easiest is to convert each of the dictionaries items to a list (which can be added together) and pass them to the `dict()` constructor.

```
dic1 = {1:10, 2:20}
dic2 = {3:30, 4:40}
dic3 = {5:50, 6:60}
nums =dict(list(dic1.items())+list(dic2.items())+list(dic3.items()))
print(nums)
```

```
main
C:\Users\bajra\PycharmProjects\pythonProject\venv\Scripts\python.exe C:\Use
{1: 10, 2: 20, 3: 30, 4: 40, 5: 50, 6: 60}

Process finished with exit code 0
```

(b) Write code to add a new key/value pair to the dictionary `nums`: `(7, 70)`

```python
dic1 = {1:10, 2:20}
dic2 = {3:30, 4:40}
dic3 = {5:50, 6:60}
nums =dict(list(dic1.items())+list(dic2.items())+list(dic3.items
print(nums)
nums.update({7:70})
print(nums)
```

```
main
C:\Users\bajra\PycharmProjects\pythonProject\venv\Scripts\python.exe (
{1: 10, 2: 20, 3: 30, 4: 40, 5: 50, 6: 60}
{1: 10, 2: 20, 3: 30, 4: 40, 5: 50, 6: 60, 7: 70}

Process finished with exit code 0
```

(c) Write code to update the value of the item with key `3` in `nums` to `80`

```python
dic1 = {1:10, 2:20}
dic2 = {3:30, 4:40}
dic3 = {5:50, 6:60}
nums =dict(list(dic1.items())+list(dic2.items())+list(dic3.items()))
nums.update({7:70})
nums[3]= 80
print(nums)
```

(d) Write code to remove the third item from dictionary `nums`.

```
dic1 = {1:10, 2:20}
dic2 = {3:30, 4:40}
dic3 = {5:50, 6:60}
nums = dict(list(dic1.items()) + list(dic2.items()) + list(dic3.items()))
nums.update({7: 70})
nums[3] = 80
del nums[3]
print(nums)
```

(e) Write code to sum all the items in the dictionary `nums`

```
dic1 = {1:10, 2:20}
dic2 = {3:30, 4:40}
dic3 = {5:50, 6:60}
nums = dict(list(dic1.items()) + list(dic2.items()) + list(dic3.items()))
nums.update({7: 70})
nums[3] = 80
del nums[3]
print(nums)
sum = sum(nums.values())
print(sum)
```

(f) Write code to multiply all the items in the dictionary `nums`

(g) Write code to retrieve the maximum and minimum values in `nums`.

```python
dic1 = {1:10, 2:20}
dic2 = {3:30, 4:40}
dic3 = {5:50, 6:60}
nums =dict(list(dic1.items())+list(dic2.items())+list(dic3.items()))
nums.update({7:70})
nums[3]= 80
del nums[3]
print(nums)
max_value = max(nums.values())
min_value= min(nums.values())
print("maximum value :", max_value)
print("minimum value:",min_value)
```

```
main
min_value: int = min(nums.values())
```

```
C:\Users\bajra\PycharmProjects\pythonProject\venv\Scripts\python.exe C:\Users\
{1: 10, 2: 20, 4: 40, 5: 50, 6: 60, 7: 70}
maximum value : 70
minimum value: 10

Process finished with exit code 0
```

3. Create a dictionary named `password_lookup` that contains usernames as keys and passwords as associated string values. Make up data for five entries.

```python
password_lookup = {
    "manogya": "12345",
    "sandesh": "54321",
    "bam": "pw",
    "sam": "123",
    "jen": "abcd"
}
print("the list of username and password :", password_lookup)
```

```
m"
n
\Users\bajra\PycharmProjects\pythonProject\venv\Scripts\python.exe C:\Users\bajra\PycharmProjects\pythonProject\main.py
e list of username and password : {'manogya': '12345', 'sandesh': '54321', 'bam': 'pw', 'sam': '123', 'jen': 'abcd'}

ocess finished with exit code 0
```

4. Write a program that creates an initially empty dictionary named `password_lookup`, prompting one-by-one for usernames and passwords (until a username of 'z' is read) entering each into the dictionary.

```python
password_lookup = {}
while True:
    username = input("enter username(or 'z' to exit):")
    if username =="z":
        break
    password = input("enter a password:")
    password_lookup[username] = password
print(password_lookup)
```

```
while True
```

```
main
C:\Users\bajra\PycharmProjects\pythonProject\venv\Scripts\python.ex
enter username(or 'z' to exit): manogya
enter a password: 123
enter username(or 'z' to exit):z
{' manogya': ' 123'}

Process finished with exit code 0
```

5. Create a dictionary named `password_hint` that contains email addresses as keys, and associated values that contain both the users' "password security question," and the answer to the question. Make up data for dictionary entries.

```python
password_lookup = {}
while True:
    username = input("enter username(or 'z' to exit):")
    if username =="z":
        break
    password = input("enter a password:")
    password_lookup[username] = password
print(password_lookup)
```

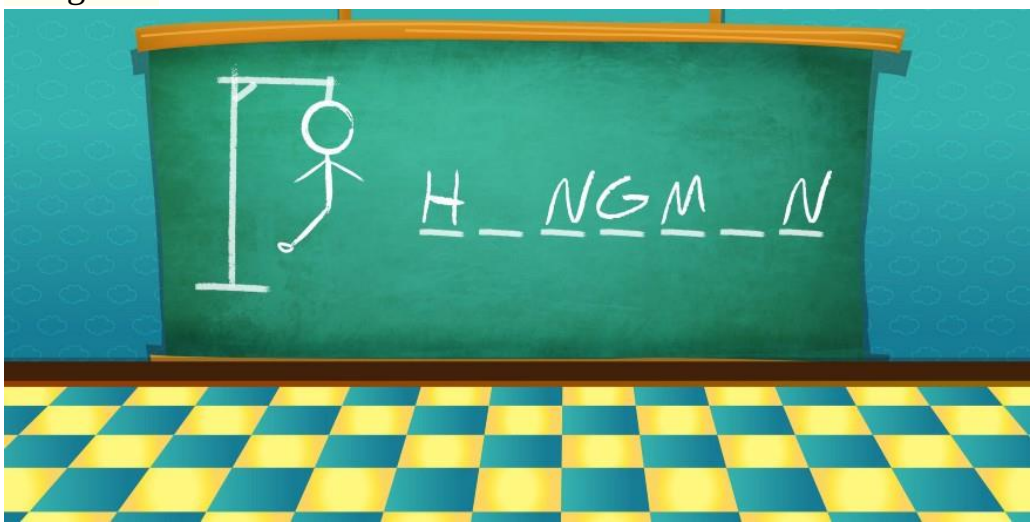
```
while True
main
C:\Users\bajra\PycharmProjects\pythonProject\venv\Scripts\python.ex
enter username(or 'z' to exit): manogya
enter a password: 123
enter username(or 'z' to exit):z
{' manogya': ' 123'}

Process finished with exit code 0
```

6. Create a dictionary named `member_table` that contains users' email addresses as keys, and answers to their password hints as the associated values, and a function that generates a temporary new password and stored in the table.

Part 4 (Home Task)

1. The hangman game introduces many new concepts like *methods*, which are functions attached to values. You'll also need to learn about a data type called

a *list*. Once you understand these concepts, it will be much easier to program Hangman.



1. You will need *random* module.
2. You will need to use the concept of *list.*

```python
import random

with open('words.txt', 'r') as f:
    words = f.readlines()

word = random.choice(words)[:-1]

allowed_guess = 7
guesses = []
done = False

while not done:
    for letter in word:
        if letter.lower() in guesses:
            print(letter, end=" ")
        else:
            print("_", end=" ")
    print("")

    guess = input(f"Allowed Errors Left {allowed_guess}, Next Guess: ")
    guesses.append(guess.lower())
    if guess.lower() not in word.lower():
        allowed_guess -= 1
        if allowed_guess == 0:
            break

    done = True
    for letter in word:
        if letter.lower() not in guesses:
            done = False

if done:
    print(f"Yoy found the word! It was {word}! :)")
else:
    print(f"Game over The word was {word}! :(")
```