

# **FP5.0 MODULE-1**

## **ASSIGNMENT**

**Batch Name:**

Infosys FP5.0 Summer 2018

**Enrollment Number:** R171217041

**SAPID:** 500060720

**Name:** Nishkarsh Raj Khare

**Sem:** Semester-III

**Branch:** CSE-DevOps-Xebia

### **Assignment-1**

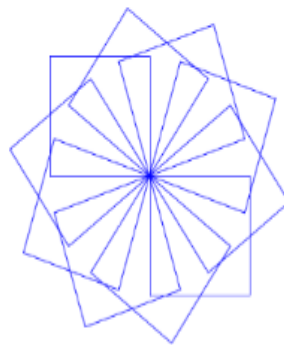
Create the following using Scratch:

1. Take a text input from the user as one of the three shape names - "square", "triangle" or "pentagon". Based on the input, draw either a red square, yellow triangle or black pentagon.

Hint - Refer the [link](#).

2. Create the following pattern using Scratch.

Hint: Create 10 squares each with an inclination of 36 degrees from the preceding square.

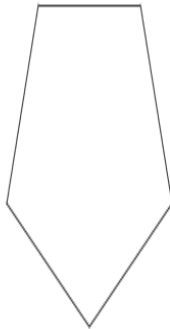


3. Create or import two Sprites and use your imagination to make them do different actions simultaneously. For example: "A bird is flying and a dog is walking at the same time."

Hint - Refer the [video](#).

a1  
v460.0.1

input 0



X: 240 Y: 157

Sprites

New sprite:



Sprite1

Stage  
1 backdrop

New backdrop:



Scripts

Costumes

Sounds

Motion

Looks

Sound

Pen

Data

Events

Control

Sensing

Operators

More Blocks

move 10 steps

turn 15 degrees

turn 15 degrees

point in direction 90°

point towards mouse-pointer

go to x: -157 y: -85

go to mouse-pointer

glide 1 secs to x: -157 y: -85

change x by 10

set x to 0

change y by 10

set y to 0

if on edge, bounce

set rotation style left-right

ask Input shape and wait

set input to answer

if then

when I receive square

set pen color to red

repeat 4

pen down

move 100 steps

turn 90 degrees

when I receive pentagon

repeat 5

set pen color to black

pen down

move 100 steps

turn 72 degrees

when I receive triangle

repeat 3

set pen color to orange

pen down

move 100 steps

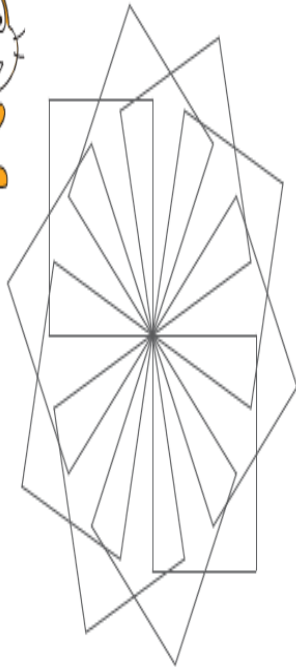
turn 120 degrees

x: -157  
y: -85



a2

v460.0.1



X: 240 Y: -180

Sprites

New sprite:



Sprite2

Stage  
1 backdrop

New backdrop:



Scripts

Costumes

Sounds

Motion

Looks

Sound

Pen

Data

Events

Control

Sensing

Operators

More Blocks

move 10 steps

turn 15 degrees

turn 15 degrees

point in direction 90°

point towards mouse-pointer

go to x: -180 y: 100

go to mouse-pointer

glide 1 secs to x: -180 y: 100

change x by 10

set x to 0

change y by 10

set y to 0

if on edge, bounce

set rotation style left-right

when clicked

repeat 10

turn 36 degrees

repeat 4

set pen color to

pen down

move 100 steps

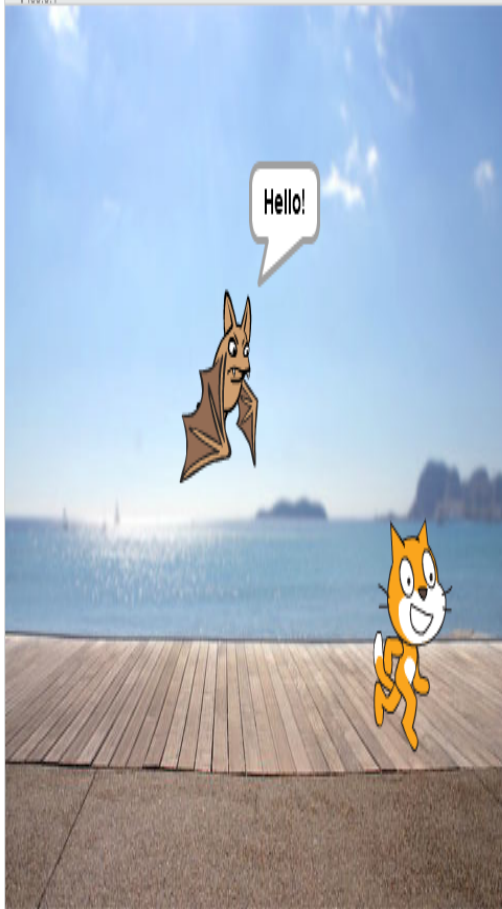
turn 90 degrees

x: -180  
y: 100



a33

v480.0.1



X: 118 Y: 93

Sprites

New sprite:

Stage  
2 backdrops

New backdrop:



Sprite1



Bat1

Scripts

Costumes

Sounds

Motion

Events

Looks

Control

Sound

Sensing

Pen

Operators

Data

More Blocks

move 10 steps

turn 15 degrees

turn 15 degrees

point in direction 90°

point towards mouse-pointer

go to x: 188 y: -70

go to mouse-pointer

glide 1 secs to x: 188 y: -70

change x by 10

set x to 0

change y by 10

set y to 0

if on edge, bounce

set rotation style left-right

when clicked

forever

wait 0.1 secs

move 10 steps

next costume

if on edge, bounce

x: 148  
y: -70

Type here to search

17:00  
15-06-2018

## **Assignment 2**

### ≡ Assignment 2

Duration : 30 mins

Create flowcharts for the following problems:

1. Calculate the average of three numbers. If average is greater than or equal to 75, print "Pass", else print "Fail".
2. Calculate and print the factorial of a number.
3. Accept the lengths of three sides of a triangle as input from the user. Based on the input, print if the given triangle is "Equilateral", "Isosceles" or "Scalene".
4. Accept the values of principal amount, rate of interest and number of years as an input from the user. Calculate and print the simple interest.

Hint - Formula for simple interest:

$$\text{Simple Interest} = (\text{principal amount} * \text{rate of interest} * \text{number of years}) / 100$$

Raptor - avg.rap

File Edit Scale View Run Mode Ink Window Generate Help

80

**Symbols**

- Assignment
- Call
- Input
- Output
- Selection
- Loop

main

Flowchart:

```
graph TD; Start([Start]) --> GetA[/"Enter 1st number"  
GET a/]; GetA --> GetB[/"Enter 2nd number"  
GET b/]; GetB --> GetC[/"Enter 3rd number"  
GET c/]; GetC --> AvgCalc[avg ← a + b + c / 3]; AvgCalc --> Decision{avg >= 75}; Decision -- Yes --> PutPass[/PUT "PASS"/]; Decision -- No --> PutFail[/PUT "FAIL"/]; PutPass --> End([End]); PutFail --> End;
```

Execution Results:

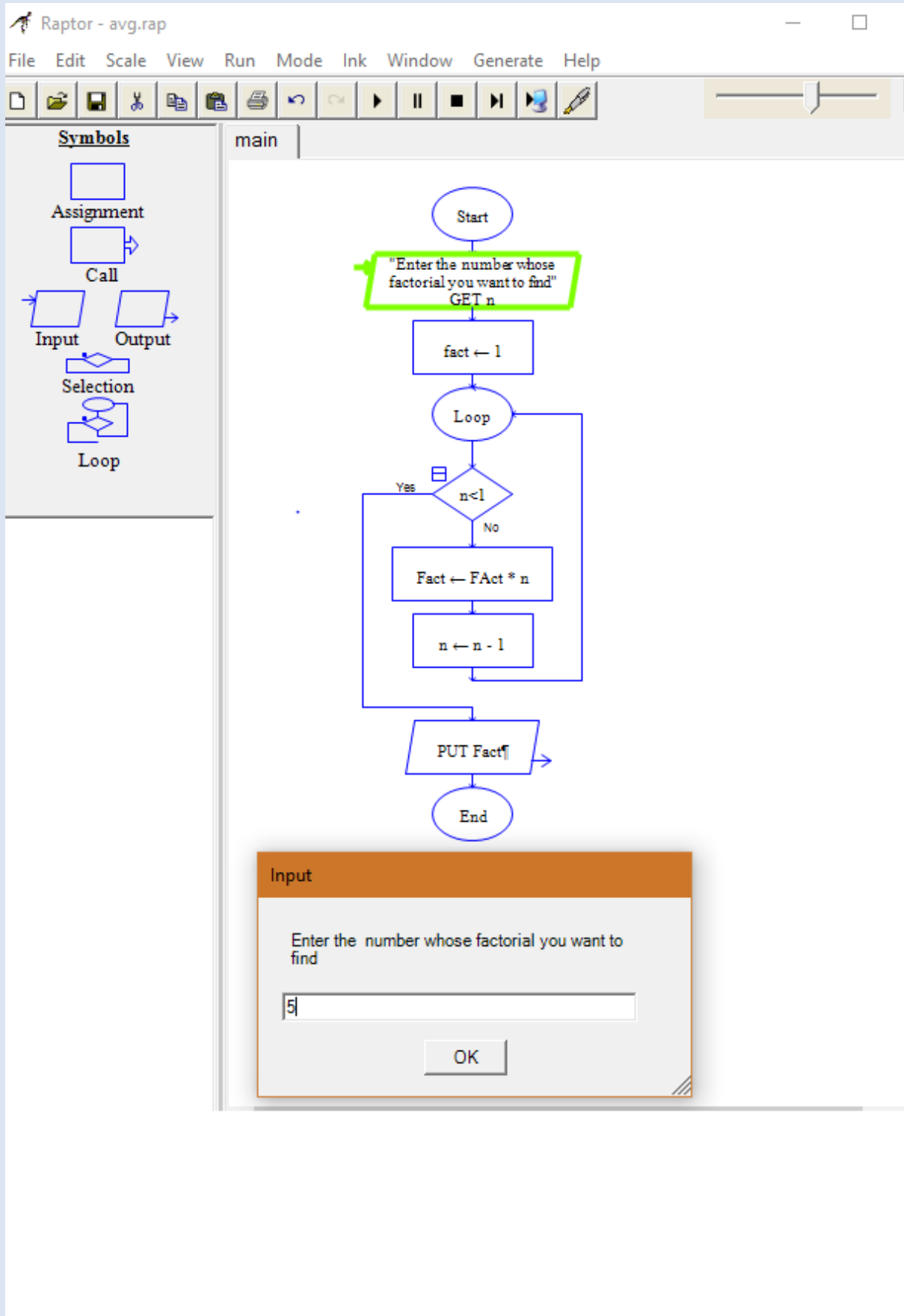
```
a: 100  
avg: 233.3333  
b: 100  
c: 100
```

MasterConsole

Font Font Size Edit Help

FAIL  
----Run complete. 8 symbols evaluated.----

PASS  
----Run complete. 8 symbols evaluated.----

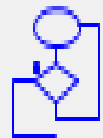




Input    Output



Selection



Loop

fact: 120

n: 0

fact ← 1



MasterConsole

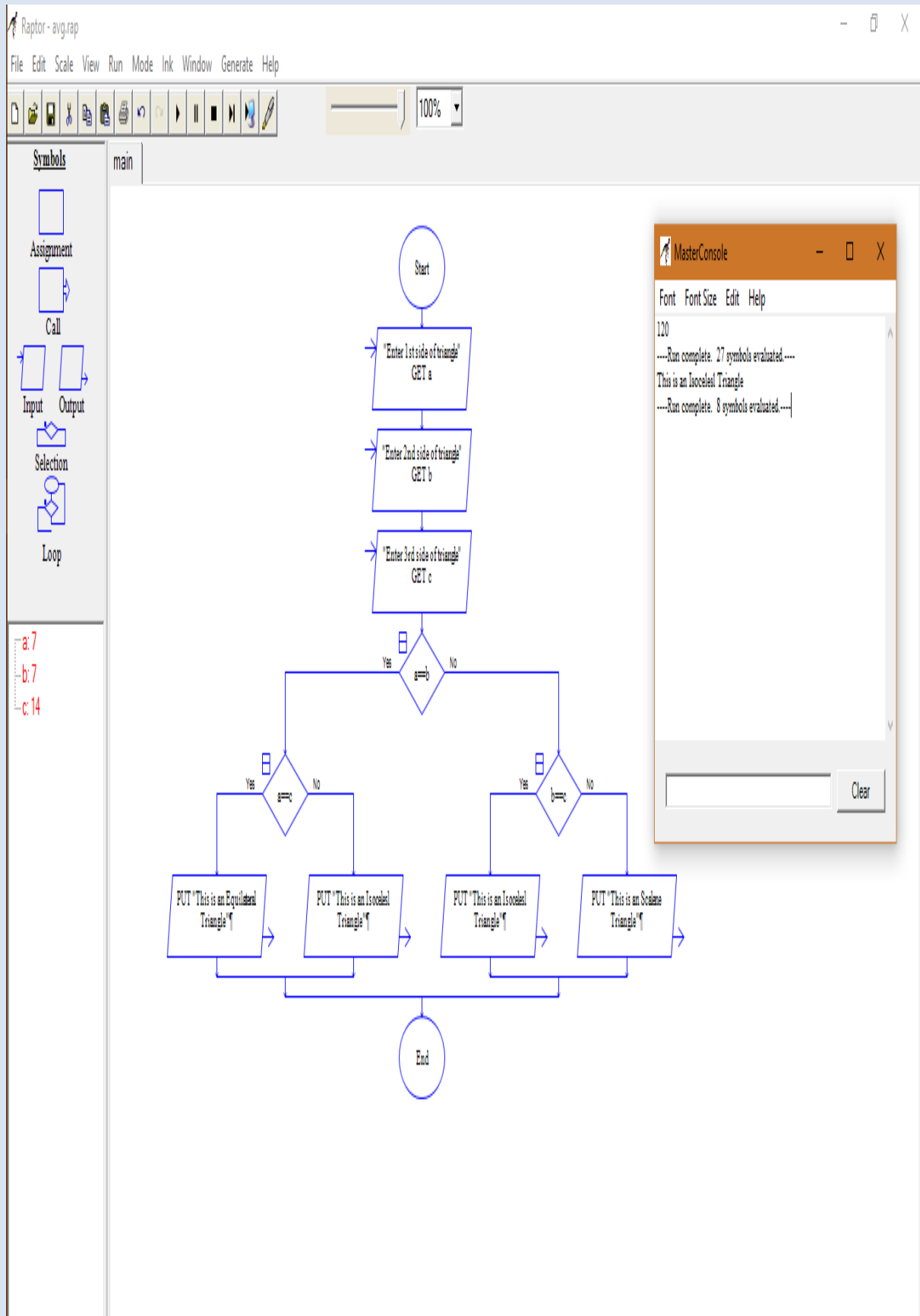


Font   Font Size   Edit   Help

120

----Run complete. 27 symbols evaluated.----

Clear



GET b

Input

Enter 1st side of triangle

7

OK

is an Isoclesl

PUT "This is an Isoclesl

PUT

GET b

Input

Enter 2nd side of triangle

7

OK

is an Isoclesl  
angle"

PUT "This is an Isoclesl  
Triangle"

PUT

Input

Enter 3rd side of triangle

14

OK

is an Isoclesl  
angle"

PUT "This is an Isoclesl  
Triangle"

PUT

Raptor - avg.rap

File Edit Scale View Run Mode Ink Window Generate Help

80%

**Symbols**

- Assignment
- Call
- Input
- Output
- Selection
- Loop

main

```
graph TD
    Start((Start)) --> GetP[/"Enter the principle"  
GET p/]
    GetP --> GetR[/"Enter the rate"  
GET r/]
    GetR --> GetT[/"Enter the time"  
GET t/]
    GetT --> SI[si ← 1]
    SI --> SI2[si ← p * r * t / 100]
    SI2 --> Put[/PUT "simple  
Intrest=" + si/]
    Put --> End((End))
```

p: 500  
r: 2  
si: 20  
t: 2

**MasterConsole**

Font Font Size Edit Help

simple Intrest=20  
----Run complete. 8 symbols evaluated.----

Clear

Input

Enter the principle

OK

Input

Enter the rate

OK

Input

Enter the time

OK

## **Assignment 3**

### Assignment 3

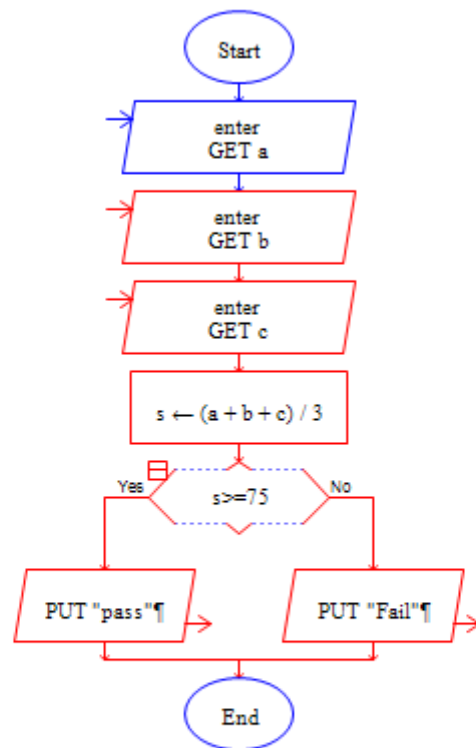
Duration : 50 mins

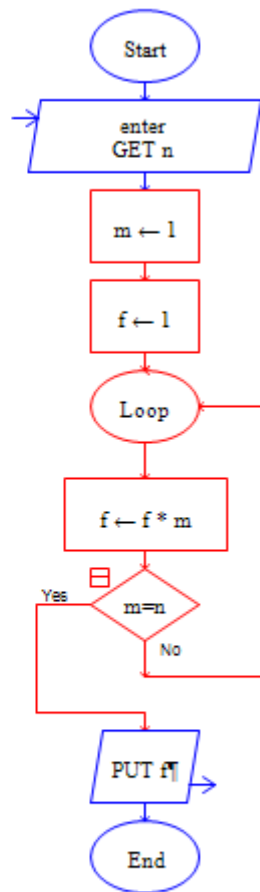
In previous section, you have created the flowcharts for the following problems. Now, use Raptor tool to create and execute flowcharts for these problems. Observe the output for different set of inputs.

- 1) Calculate the average of three numbers. If average is greater than or equal to 75, print "Pass", else print "Fail".
- 2) Calculate and print the factorial of a number.
- 3) Accept the lengths of three sides of a triangle as input from the user. Based on the input, print if the given triangle is "Equilateral", "Isosceles" or "Scalene".
- 4) Accept the values of principal amount, rate of interest and number of years as an input from the user. Calculate and print the simple interest.

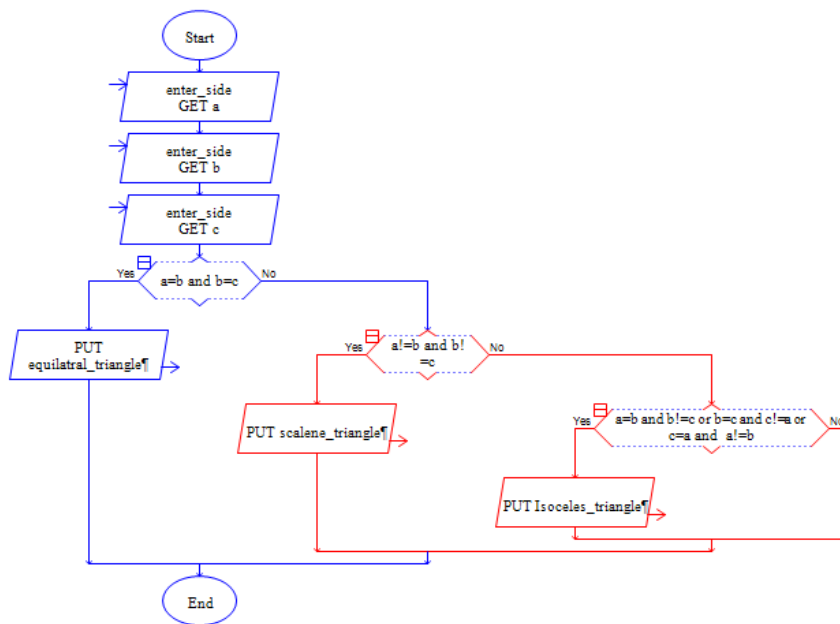
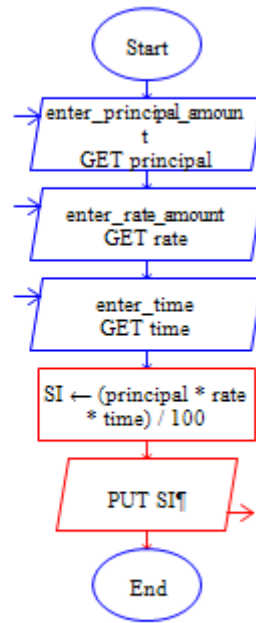
Hint - Formula for simple interest:

$$\text{Simple Interest} = (\text{principal amount} * \text{rate of interest} * \text{number of years}) / 100$$









## **Assignment 4**

### Assignment 4

Write Pseudo Code:

1. to check whether a given number is even or odd.
2. to print the multiples of 3 between 1 to 20.
3. to find factorial of a given number.
4. to calculate 'x' to the power of 'n' using a while loop.
  - Assume both 'x' and 'n' are positive whole numbers.

**1)**

**1) Start**

**2) Input number as a variable**

**3) If remainder of number/2 is =0**

**Print even**

**else**

**print Odd**

**4) End**

**2)**

**1. Start**

**2. k=1**

**3. While p<20**

**4. P=k\*3**

**5. Print p**

**6.  $k=k+1$**

**7. End**

**3)**

**1. Start**

**2. Take n as an integer to find its factorial**

**3.  $p=1$**

**4. While  $n>0$**

**5.  $P=p*n$**

**6. End**

**4)**

**1. Start**

**2. Input values of x and n**

**3.  $P=1$**

**4. While  $p \leq n$**

**5.  $x=x*x$**

**6. End**

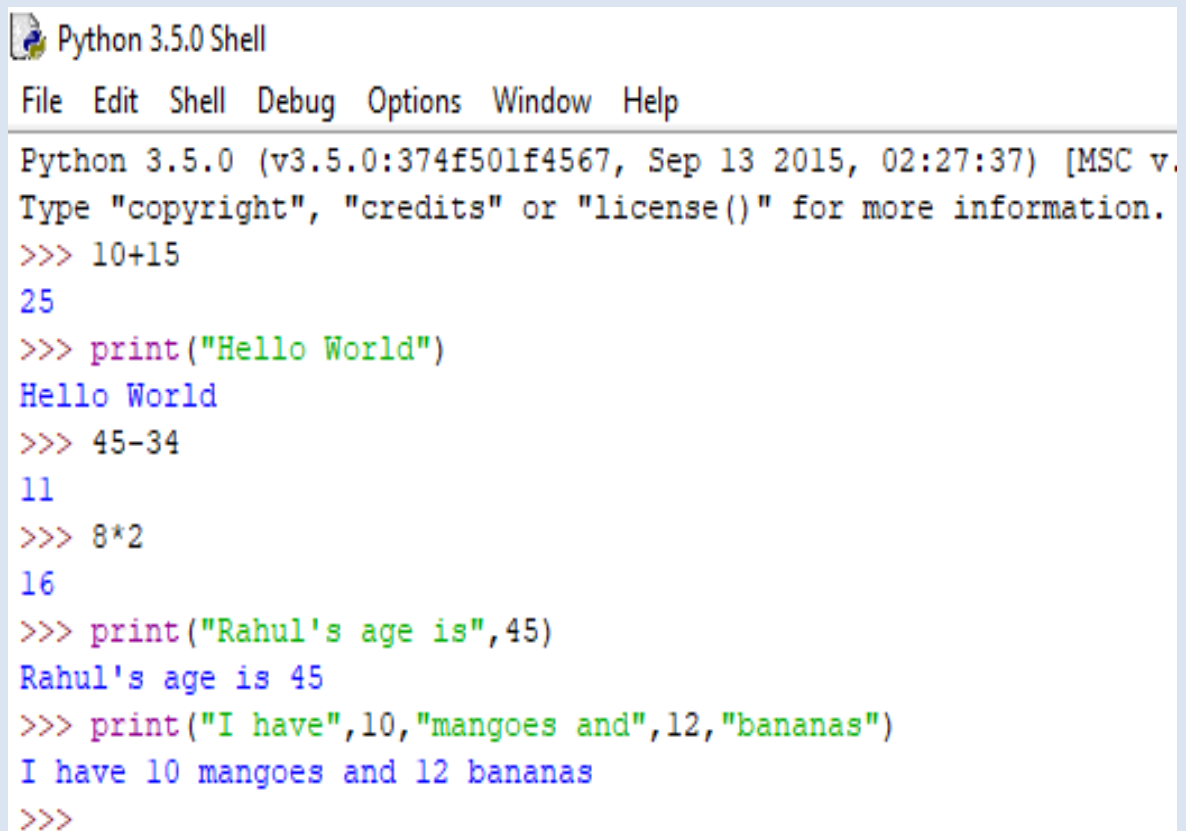
## **Assignment 5**

### Assignment 5

Duration : 10 mins

Open the Python IDLE and execute the following commands. Observe the output.

- 1) `10 + 15`
- 2) `print("Hello World")`
- 3) `45 - 34`
- 4) `8 * 2`
- 5) `print("Rahul's age is", 45)`
- 6) `print("I have", 10, "mangoes and", 12, "bananas")`



```
Python 3.5.0 Shell
File Edit Shell Debug Options Window Help
Python 3.5.0 (v3.5.0:374f501f4567, Sep 13 2015, 02:27:37) [MSC v.
Type "copyright", "credits" or "license()" for more information.
>>> 10+15
25
>>> print("Hello World")
Hello World
>>> 45-34
11
>>> 8*2
16
>>> print("Rahul's age is",45)
Rahul's age is 45
>>> print("I have",10,"mangoes and",12,"bananas")
I have 10 mangoes and 12 bananas
>>>
```

## **Assignment 6**

### Assignment 6

Open Python IDLE and execute the following commands. Observe the output.

1. `emp_number = 1233`
2. `print("Employee Number:", emp_number)`
3. `emp_salary = 16745.50`
4. `emp_name = "Jerry Squaris"`
5. `print("Employee Salary and Name:", emp_salary, emp_name)`
6. `emp_salary = 23450.34`
7. `print("Updated Employee Salary:", emp_salary)`

#### Python 3.5.0 Shell

File Edit Shell Debug Options Window Help

Python 3.5.0 (v3.5.0:374f501f4567, Sep 13 2015, 02:27:37) [MSC v.1400 64-bit (AMD64)]  
Type "copyright", "credits" or "license()" for more information.

```
>>> emp_number=1233
```

```
>>> print("Employee Number:",emp_number)
```

```
Employee Number: 1233
```

```
>>> emp_salary=16745.50
```

```
>>> emp_name="Jerry Squaris"
```

```
>>> print("Employee Salary and Name:",emp_salary,emp_name)
```

```
Employee Salary and Name: 16745.5 Jerry Squaris
```

```
>>> emp_salary=23450.34
```

```
>>> print("Updated Employee Salary:",emp_salary)
```

```
Updated Employee Salary: 23450.34
```

```
>>>
```

## **Assignment 7**

### Assignment 7

Execute the following Python statements in IDLE and observe the output:

- `customer_id = 101`
- `type(customer_id)`
- `customer_name = "John"`
- `type(customer_name)`
- `bill_amount = 675.45`
- `type(bill_amount )`
- `x = 5.3 + 0.9j`
- `type(x)`
- `print(customer_id, customer_name, bill_amount)`
- `print(x.real)`
- `print(x.imag + 3)`
- `Flag = True`
- `type(Flag)`
- `y = "False"`

assignment7.py - C:\Users\500060720\Desktop\Post Freshmann\Locker

File Edit Format Run Options Window Help

```
customer_id=101
type(customer_id)
customer_name="John"
type(customer_name)
bill_amount=675.45
type(bill_amount)
x=5.3+0.9j
type(x)
print(customer_id,customer_name,bill_amount)
print(x.real)
print(x.imag+3)
Flag=True
type(Flag)
y="False"
```

Python 3.5.0 Shell

File Edit Shell Debug Options Window Help

Python 3.5.0 (v3.5.0:374f501f4567, Sep 13 2015, 02:27:37) [MSC v.1900 64 bit (AMD64)] on win32  
Type "copyright", "credits" or "license()" for more information.

>>>

RESTART: C:\Users\500060720\Desktop\Post Freshmann\Locker\Python Project\assignment7.py

101 John 675.45

5.3

3.9

>>>

## **Assignment 8**

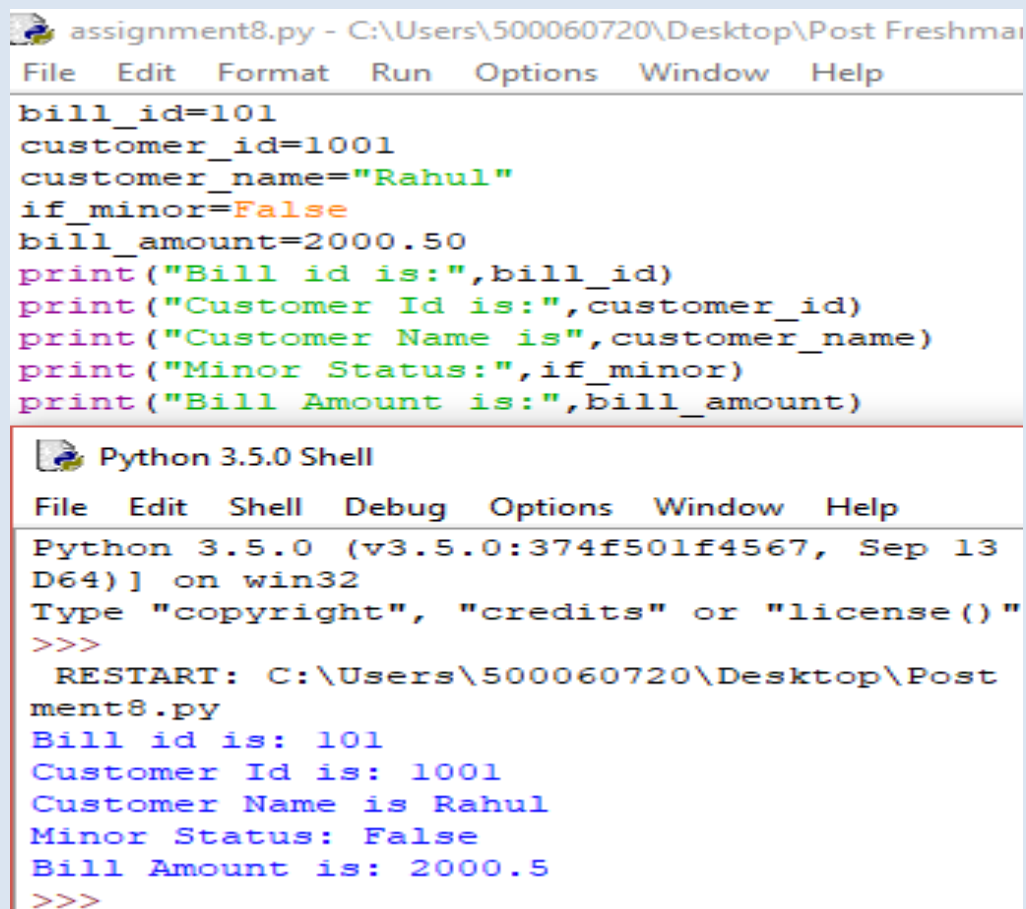
### Assignment 8

Duration : 10 mins

In a retail application, shopkeeper wants to keep a track of following details of a customer. Sample values are provided.

- bill\_id = 101
- customer\_id = 1001
- customer\_name = "Rahul"
- if\_minor = False
- bill\_amount = 2000.50

Write a python program to store the details and display them.



```
assignment8.py - C:\Users\500060720\Desktop\Post Freshman
File Edit Format Run Options Window Help
bill_id=101
customer_id=1001
customer_name="Rahul"
if_minor=False
bill_amount=2000.50
print("Bill id is:",bill_id)
print("Customer Id is:",customer_id)
print("Customer Name is",customer_name)
print("Minor Status:",if_minor)
print("Bill Amount is:",bill_amount)

Python 3.5.0 Shell
File Edit Shell Debug Options Window Help
Python 3.5.0 (v3.5.0:374f501f4567, Sep 13
D64) ] on win32
Type "copyright", "credits" or "license()"
>>>
RESTART: C:\Users\500060720\Desktop\Post
ment8.py
Bill id is: 101
Customer Id is: 1001
Customer Name is Rahul
Minor Status: False
Bill Amount is: 2000.5
>>>
```



## Assignment 9

### Assignment 9

Duration : 5 mins

Execute the following commands and observe the usage of different types of commenting styles.

```
i = 10    # creates an integer variable. This is a single line comment.
```

```
print("i =", i)    # prints 10
```

```
...
```

Below code creates a Boolean variable in Python

(This is a multiple line comment)

```
...
```

```
s = True
```

```
print("s =", s)    #prints True, Here, s is a Boolean variable with value True
```

```
....
```

Below code assigns string data to variable 's'. Data type of variable can change during execution,

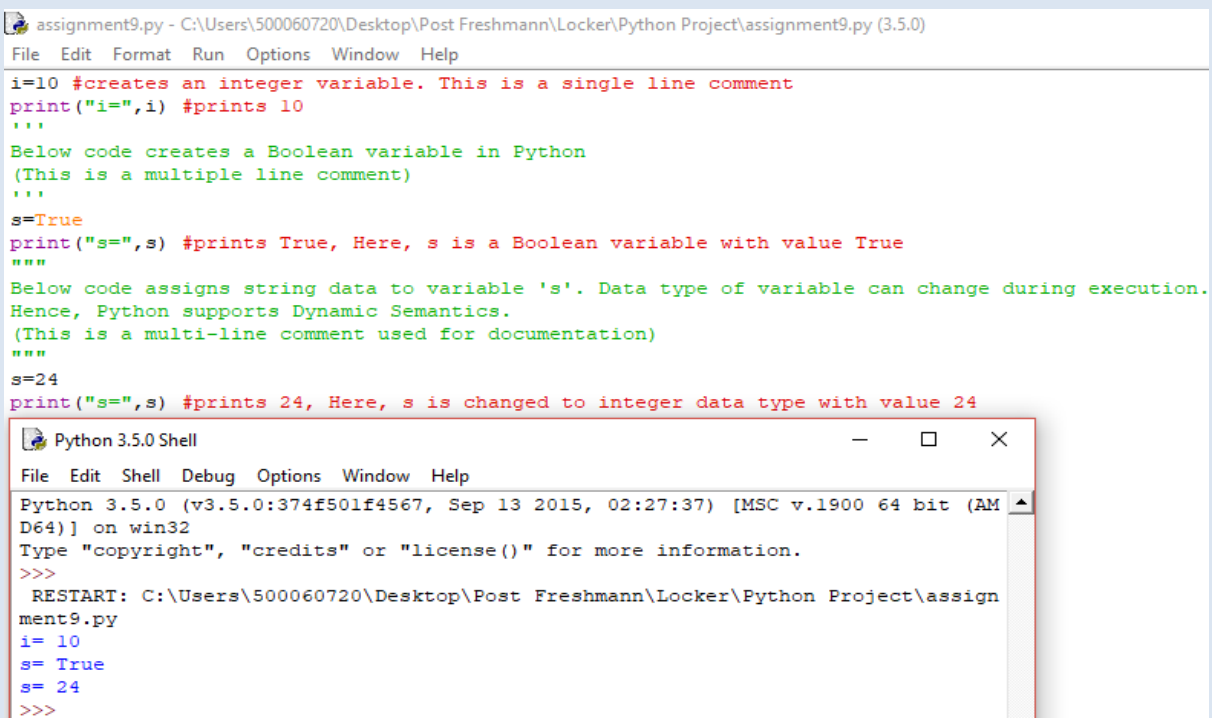
Hence, Python supports Dynamic Semantics.

(This is multi-line comment used for documentation)

```
....
```

```
s = 24
```

```
print("s =", s)    #prints 24, Here, s is changed to integer data type with value 24
```



The screenshot shows a Python IDE window titled 'assignment9.py - C:\Users\500060720\Desktop\Post Freshmann\Locker\Python Project\assignment9.py (3.5.0)'. The code in the editor is as follows:

```
File Edit Format Run Options Window Help
i=10 #creates an integer variable. This is a single line comment
print("i=",i) #prints 10
...
Below code creates a Boolean variable in Python
(This is a multiple line comment)
...
s=True
print("s=",s) #prints True, Here, s is a Boolean variable with value True
....
Below code assigns string data to variable 's'. Data type of variable can change during execution.
Hence, Python supports Dynamic Semantics.
(This is a multi-line comment used for documentation)
....
s=24
print("s=",s) #prints 24, Here, s is changed to integer data type with value 24
```

Below the code editor is a 'Python 3.5.0 Shell' window showing the execution output:

```
Python 3.5.0 Shell
File Edit Shell Debug Options Window Help
Python 3.5.0 (v3.5.0:374f501f4567, Sep 13 2015, 02:27:37) [MSC v.1900 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:\Users\500060720\Desktop\Post Freshmann\Locker\Python Project\assignment9.py
i= 10
s= True
s= 24
>>>
```

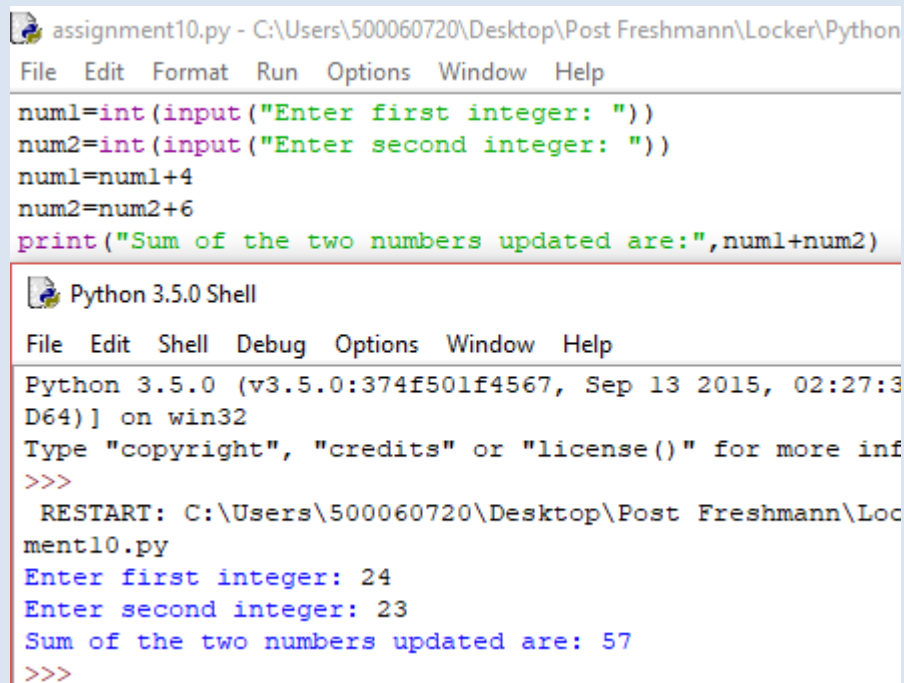
## **Assignment 10**

### Assignment 10

Write a Python program for the following requirements:

- Prompt the user to input two numbers num1 and num2
- Increment num1 by 4 and num2 by 6
- Find and print the sum of new values of num1 and num2

Hint - Use type casting for converting the input into an integer.



```
assignment10.py - C:\Users\500060720\Desktop\Post Freshmann\Locker\Python
File Edit Format Run Options Window Help
num1=int(input("Enter first integer: "))
num2=int(input("Enter second integer: "))
num1=num1+4
num2=num2+6
print("Sum of the two numbers updated are:",num1+num2)

Python 3.5.0 Shell
File Edit Shell Debug Options Window Help
Python 3.5.0 (v3.5.0:374f501f4567, Sep 13 2015, 02:27:3
D64)] on win32
Type "copyright", "credits" or "license()" for more inf
>>>
RESTART: C:\Users\500060720\Desktop\Post Freshmann\Loc
ment10.py
Enter first integer: 24
Enter second integer: 23
Sum of the two numbers updated are: 57
>>>
```

## **Assignment 11**

### Assignment 11

Duration : 40 mins

1) Consider two variables 'a' and 'b' in Python such that  $a = 4$  and  $b = 5$ . Swap the values of 'a' and 'b' without using a temporary variable. Print the values of 'a' and 'b' before and after swapping.

2) Consider the scenario of processing marks of a student in ABC Training Institute. John, the student of fifth grade takes exams in three different subjects. Create three variables to store the marks obtained by John in three subjects. Find and display the average marks scored by John.

Now change the marks in one of the subjects and observe the output. Did the value of average change?

3) Given the value of radius of a circle, write a Python program to calculate the area and perimeter of the circle. Display both the values.

4) The finance department of a company wants to compute the monthly pay of its employees. Monthly pay should be calculated as mentioned in the formula below. Display all the employee details.

Monthly Pay = Number of hours worked in a week \* Pay rate per hour \* No. of weeks in a month


- The number of hours worked by the employee in a week should be considered as 40
- Pay rate per hour should be considered as Rs.400
- Number of weeks in a month should be considered as 4

Write a Python program to implement the above real world problem.

 assignment11\_1.py - C:\Users\500060720\Desktop\Post Freshmann\Locker\Pytl

File Edit Format Run Options Window Help

```
a,b=4,5
print("Value of a and b before swapping are:",a,b)
a=a+b;
b=a-b;
a=a-b;
print("Values of a and b after swapping are:",a,b)
```

 Python 3.5.0 Shell

File Edit Shell Debug Options Window Help

Python 3.5.0 (v3.5.0:374f501f4567, Sep 13 2015, 02:27: D64)] on win32

Type "copyright", "credits" or "license()" for more in  
>>>

RESTART: C:\Users\500060720\Desktop\Post Freshmann\Lo  
ment11\_1.py

Value of a and b before swapping are: 4 5

Values of a and b after swapping are: 5 4

>>>

assignment11\_2.py - C:\Users\500060720\Desktop\Post Freshmann\Locker\Python Project\assignment11\_2.py (3.5.0)

File Edit Format Run Options Window Help

```
Institute="ABC"
Student_name="John"
grade="fifth"
a=int(input("Enter Marks in Subject 1: "))
b=int(input("Enter Marks in Subject 2: "))
c=int(input("Enter Marks in Subject 3: "))
average=(a+b+c)/3;
print("Marks of",Student_name,"of",Institute,"Institue,studying in grade",grade,"are")
print("Marks in subject 1 are:",a)
print("Marks in subject 2 are:",b)
print("Marks in subject 3 are:",c)
print("Average of the marks are:",average)
a=int(input("Re-enter Marks in Subject 1: "))
average=(a+b+c)/3;
print("Average of the marks are:",average)
```

Python 3.5.0 Shell

File Edit Shell Debug Options Window Help

Python 3.5.0 (v3.5.0:374f501f4567, Sep 13 2015, 02:27:37) [MSC v.1900 64 bit (AMD64)] on win32

Type "copyright", "credits" or "license()" for more information.

>>>

RESTART: C:\Users\500060720\Desktop\Post Freshmann\Locker\Python Project\assignment11\_2.py

Enter Marks in Subject 1: 24

Enter Marks in Subject 2: 29

Enter Marks in Subject 3: 25

Marks of John of ABC Institue,studying in grade fifth are

Marks in subject 1 are: 24

Marks in subject 2 are: 29

Marks in subject 3 are: 25

Average of the marks are: 26.0

Re-enter Marks in Subject 1: 25

Average of the marks are: 26.333333333333332

>>>

assignment11\_3.py - C:\Users\500060720\Desktop\Post Freshmann\Locker\Python Project\assignment11\_3.py (3.5.0)

File Edit Format Run Options Window Help

```
print("Program about area and perimeter of circle")
radii=float(input("Enter the radius of the circle: "))
perimeter=2*3.14*radii
area=3.14*radii*radii
print("Radii of the circle is",radii,"and it has perimeter=",perimeter,"and area=",area)
```

Python 3.5.0 Shell

File Edit Shell Debug Options Window Help

Python 3.5.0 (v3.5.0:374f501f4567, Sep 13 2015, 02:27:37) [MSC v.1900 64 bit (AMD64)] on win32

Type "copyright", "credits" or "license()" for more information.

>>>

RESTART: C:\Users\500060720\Desktop\Post Freshmann\Locker\Python Project\assignment11\_3.py

Program about area and perimeter of circle

Enter the radius of the circle: 52

Radii of the circle is 52.0 and it has perimeter= 326.56 and area= 8490.56

>>>

assignment11\_4.py - C:\Users\500060720\Desktop\Post Freshmann\Locker\Python Project\assignment11\_4.py (3.5.0)

File Edit Format Run Options Window Help

```
print("Company Financial Status of Employee")
name="Nishkarsh Raj Khare"
age=10
pay_per_hour=400
number_hours_worked=40
no_of_weeks=4
monthly_pay=pay_per_hour*number_hours_worked*no_of_weeks
print("Name of the person is",name,"Age of the person is",age,"Monthly Pay of the person is",monthly_pay)
```

Python 3.5.0 Shell

File Edit Shell Debug Options Window Help

Python 3.5.0 (v3.5.0:374f501f4567, Sep 13 2015, 02:27:37) [MSC v.1900 64 bit (AMD64)] on win32

Type "copyright", "credits" or "license()" for more information.

>>>

RESTART: C:\Users\500060720\Desktop\Post Freshmann\Locker\Python Project\assignment11\_4.py

Company Financial Status of Employee

Name of the person is Nishkarsh Raj Khare Age of the person is 10 Monthly Pay of the person is 64000

>>>

## **Assignment 12**

### Assignment 12

Duration : 10 mins

Identify the sections of the given program where the coding standards are not followed and correct them.

```
itemNo=1005
```

```
unitprice = 250
```

```
quantity = 2
```

```
amount=quantity*unitprice
```

```
print("Item No:", itemNo)
```

```
print("Bill Amount:", amount)
```

**itemNo = 1005 #space before and after ‘=’**

**amount = quantity \* unitprice #spacing problem**

## **Assignment**

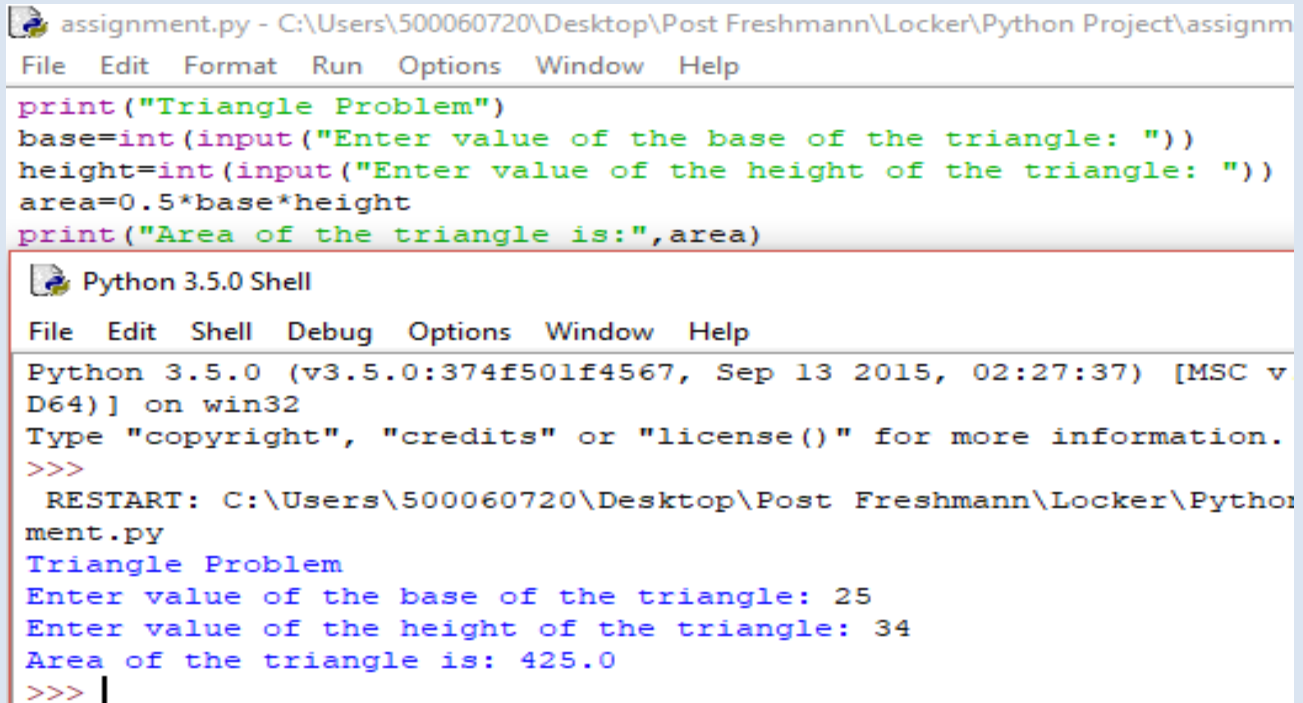
### Assignment

Duration : 20 mins

- Create a new file in Python IDLE as "triangle.py"
- Write a Python program to calculate and print the area of the triangle. Prompt the user to input the values for base and height of the triangle.
- Execute the program (use 'Run Module' under 'Run' tab) and observe the output.
- Close the file, open it again and execute it once more with different values. Observe the output.

Hint - Use type casting for converting the input into an integer.

Area of a triangle =  $\frac{1}{2} * \text{base} * \text{height}$



```
assignment.py - C:\Users\500060720\Desktop\Post Freshmann\Locker\Python Project\assignment.py
File Edit Format Run Options Window Help

print("Triangle Problem")
base=int(input("Enter value of the base of the triangle: "))
height=int(input("Enter value of the height of the triangle: "))
area=0.5*base*height
print("Area of the triangle is:",area)

Python 3.5.0 Shell
File Edit Shell Debug Options Window Help
Python 3.5.0 (v3.5.0:374f501f4567, Sep 13 2015, 02:27:37) [MSC v150032 D64] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:\Users\500060720\Desktop\Post Freshmann\Locker\Python Project\assignment.py
Triangle Problem
Enter value of the base of the triangle: 25
Enter value of the height of the triangle: 34
Area of the triangle is: 425.0
>>> |
```



## **Assignment 13**

### Assignment 13

Duration : 20 mins

- 1) Consider the scenario of retail store management again. The store provides discount for all bill amounts based on the criteria below:

Bill Amount	Discount %
$\geq 1000$	5
$\geq 500$ and $< 1000$	2
$> 0$ and $< 500$	1

Write a Python program to find the net bill amount after discount. Observe the output with different values of bill amount.

Assume that bill amount will be always greater than zero.

- 2) Extend the above program to validate the customer id. Customer ids in the range of 101 and 1000 (both inclusive) should only be considered valid.

Note: Display appropriate error messages wherever applicable.

assignment13\_1.py - C:\Users\500060720\Desktop\Post Freshmann\Locker\Python Project\assignment13\_1.py

File Edit Format Run Options Window Help

```
print("Retail Store Management Bill")
bill_amount=float(input("Enter the bill amount: "))
flag=0
if (bill_amount>=1000):
    discount = 0.05*bill_amount
elif (bill_amount>=500 and bill_amount<1000):
    discount = 0.02*bill_amount
elif (bill_amount>0 and bill_amount<500):
    discount = 0.01*bill_amount
else:
    flag=1
    print("Please enter positive bill amount")
if flag==0:
    net_amount = bill_amount-discount
    print("The total bill amount after discount is:",net_amount)
print("Thank You for shopping!!!")
```

Python 3.5.0 Shell

File Edit Shell Debug Options Window Help

Python 3.5.0 (v3.5.0:374f501f4567, Sep 13 2015, 02:27:37) [MSC v.1900 64-bit] on win32

Type "copyright", "credits" or "license()" for more information.

>>>

RESTART: C:\Users\500060720\Desktop\Post Freshmann\Locker\Python Project\assignment13\_1.py

Retail Store Management Bill

Enter the bill amount: 325

The total bill amount after discount is: 321.75

Thank You for shopping!!!

>>>

assignment13\_2.py - C:\Users\500060720\Desktop\Post Freshmann\Locker\Python Project\assignment13\_2.py (3.5.0)

File Edit Format Run Options Window Help

```
print("Retail Store Management Bill")
customer_id=int(input("Hello! Please enter your Customer Id to continue: "))
if (customer_id>=101 and customer_id<=1000):
    bill_amount=float(input("Enter the bill amount: "))
    flag=0
    if (bill_amount>=1000):
        discount = 0.05*bill_amount
    elif (bill_amount>=500 and bill_amount<1000):
        discount = 0.02*bill_amount
    elif (bill_amount>0 and bill_amount<500):
        discount = 0.01*bill_amount
    else:
        flag=1
        print("Please enter positive bill amount")
    if flag==0:
        net_amount = bill_amount-discount
        print("The total bill amount after discount is:",net_amount)
    print("Thank You for shopping!!!")
else:
    print("Sorry! Your Account does not exist!!!")
```

Python 3.5.0 Shell

File Edit Shell Debug Options Window Help

Python 3.5.0 (v3.5.0:374f501f4567, Sep 13 2015, 02:27:37) [MSC v.1900 64 bit (AMD64)] on win32

Type "copyright", "credits" or "license()" for more information.

>>>

RESTART: C:\Users\500060720\Desktop\Post Freshmann\Locker\Python Project\assignment13\_2.py

Retail Store Management Bill

Hello! Please enter your Customer Id to continue: 141

Enter the bill amount: 532456

The total bill amount after discount is: 505833.2

Thank You for shopping!!!

>>>

## **Assignment 14**

### Assignment 14

Duration : 40 mins

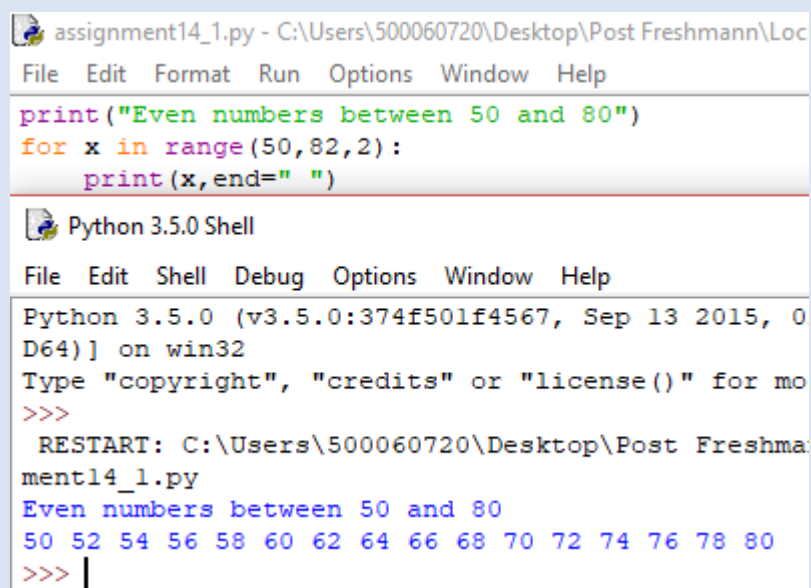
Implement the following in Python:

- Display all even numbers between 50 and 80 (both inclusive) using "for" loop.
- Add natural numbers up to n where n is taken as an input from user. Print the sum.
- Prompt the user to enter a number. Print whether the number is prime or not.
- Print Fibonacci series till nth term where n is taken as an input from user.

Hint – Fibonacci series is a series of numbers in which each number is the sum of the two preceding numbers.

Series start from 1 and goes like : 1, 1, 2, 3, 5, 8, 13 ....

Google it for further information.



```
assignment14_1.py - C:\Users\500060720\Desktop\Post Freshmann\Loc
File Edit Format Run Options Window Help
print("Even numbers between 50 and 80")
for x in range(50,82,2):
    print(x,end=" ")

Python 3.5.0 Shell
File Edit Shell Debug Options Window Help
Python 3.5.0 (v3.5.0:374f501f4567, Sep 13 2015, 0
D64)] on win32
Type "copyright", "credits" or "license()" for mo
>>>
RESTART: C:\Users\500060720\Desktop\Post Freshma
ment14_1.py
Even numbers between 50 and 80
50 52 54 56 58 60 62 64 66 68 70 72 74 76 78 80
>>> |
```

```
assignment14_2.py - C:\Users\500060720\Desktop\Post Freshmann\Locker\Python Project
File Edit Format Run Options Window Help

print("Sum of n natural numbers")
num=int(input("Enter an integer: "))
sum=0
if(num>0):
    for x in range (1,num+1):
        sum=sum+x
    print(("Sum of natural numbers till %d is %d")%(num,sum))
else:
    print("You entered non-natural number")

Python 3.5.0 Shell
File Edit Shell Debug Options Window Help
Python 3.5.0 (v3.5.0:374f501f4567, Sep 13 2015, 02:27:37) [MS
D64)] on win32
Type "copyright", "credits" or "license()" for more informati
>>>
RESTART: C:\Users\500060720\Desktop\Post Freshmann\Locker\Py
ment14_2.py
Sum of n natural numbers
Enter an integer: 36
Sum of natural numbers till 36 is 666
>>>
```

```
assignment14_3.py - C:\Users\500060720\Desktop\Post Fr
File Edit Format Run Options Window Help

print("Prime number programme!!!")
num=int(input("Enter a number: "))
flag=0
for x in range(1,num+1):
    if num%x==0:
        flag=flag+1
    else:
        pass
if(flag==2):
    print("Number is prime")
else:
    print ("Not prime")

Python 3.5.0 Shell
File Edit Shell Debug Options Window Help
Python 3.5.0 (v3.5.0:374f501f4567, Sep 13 2015, 02:27:37) [MS
D64)] on win32
Type "copyright", "credits" or "license()" for more informati
>>>
RESTART: C:\Users\500060720\Desktop\Post Freshmann\Locker\Py
ment14_3.py
Prime number programme!!!
Enter a number: 325
Not prime
>>>
```

assignment14\_4.py - C:\Users\500060720\Desktop\Post Freshmann

File Edit Format Run Options Window Help

```
print("Fibonacci Series till nth term")
n=int(input("Enter a number to set range: "))
a,b=0,1
print(b,end=" ")
for x in range(1,n-1):
    c=a+b
    print(c,end=" ")
    a=b
    b=c
print(" ")
print("Thanks for your time")
```

Python 3.5.0 Shell

File Edit Shell Debug Options Window Help

Python 3.5.0 (v3.5.0:374f501f4567, Sep 13 2015; D64) on win32

Type "copyright", "credits" or "license()" for more  
>>>

RESTART: C:\Users\500060720\Desktop\Post Freshmann\assignment14\_4.py

Fibonacci Series till nth term

Enter a number to set range: 5

1 1 2 3

Thanks for your time

>>> |

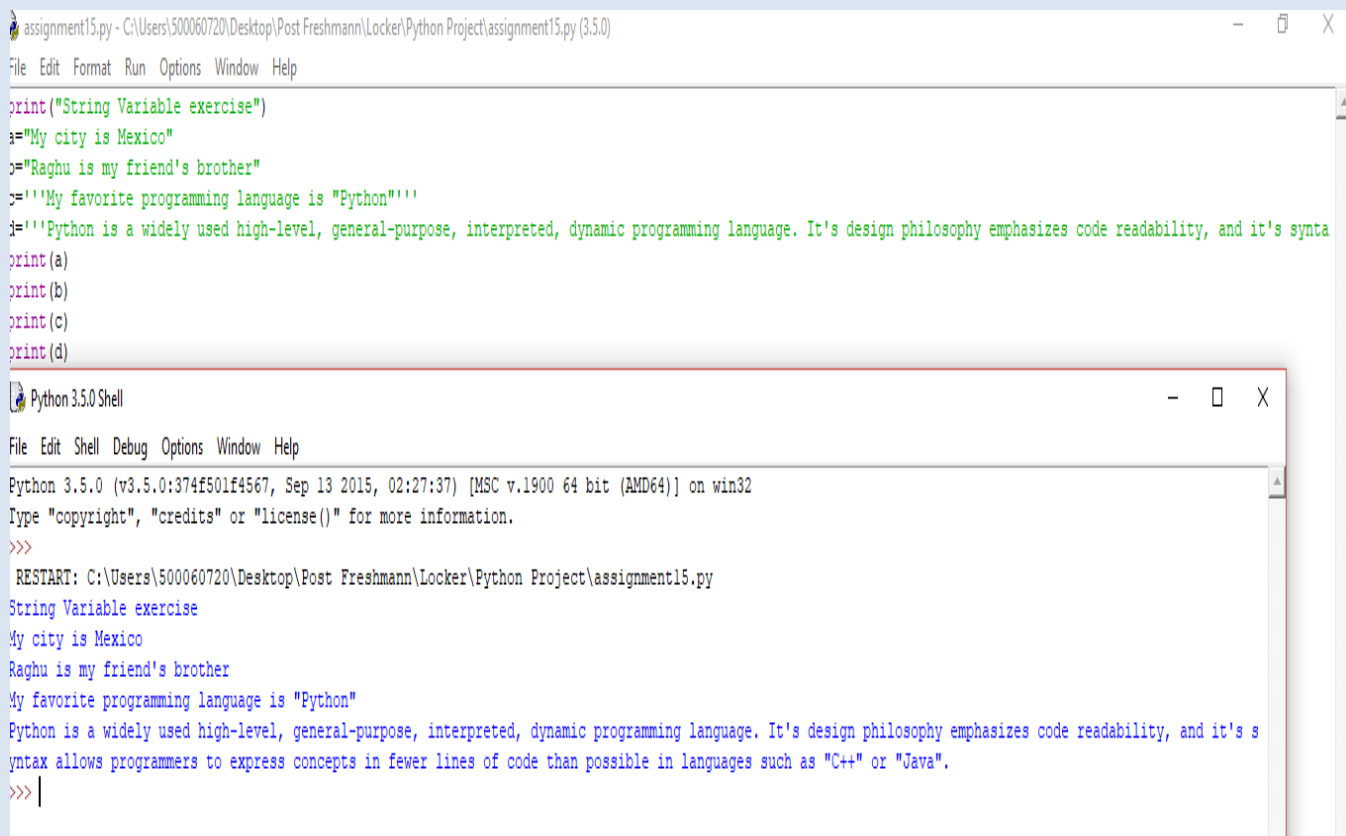
## **Assignment 15**

### Assignment 15

Duration : 15 mins

Create four string variables a, b, c, d to store the following values and display them:

- My city is Mexico
- Raghu is my friend's brother
- My favorite programming language is "Python"
- Python is a widely used high-level, general-purpose, interpreted, dynamic programming language. It's design philosophy emphasizes code readability, and it's syntax allows programmers to express concepts in fewer lines of code than possible in languages such as "C++" or "Java".



```
assignment15.py - C:\Users\500060720\Desktop\Post Freshmann\Locker\Python Project\assignment15.py (3.5.0)
File Edit Format Run Options Window Help

print("String Variable exercise")
a="My city is Mexico"
b="Raghu is my friend's brother"
c="My favorite programming language is \"Python\""
d="Python is a widely used high-level, general-purpose, interpreted, dynamic programming language. It's design philosophy emphasizes code readability, and it's syntax allows programmers to express concepts in fewer lines of code than possible in languages such as \"C++\" or \"Java\"."
print(a)
print(b)
print(c)
print(d)

Python 3.5.0 Shell
File Edit Shell Debug Options Window Help

Python 3.5.0 (v3.5.0:374f501f4567, Sep 13 2015, 02:27:37) [MSC v.1900 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:\Users\500060720\Desktop\Post Freshmann\Locker\Python Project\assignment15.py
String Variable exercise
My city is Mexico
Raghu is my friend's brother
My favorite programming language is "Python"
Python is a widely used high-level, general-purpose, interpreted, dynamic programming language. It's design philosophy emphasizes code readability, and it's syntax allows programmers to express concepts in fewer lines of code than possible in languages such as "C++" or "Java".
>>> |
```

## **Assignment 16**

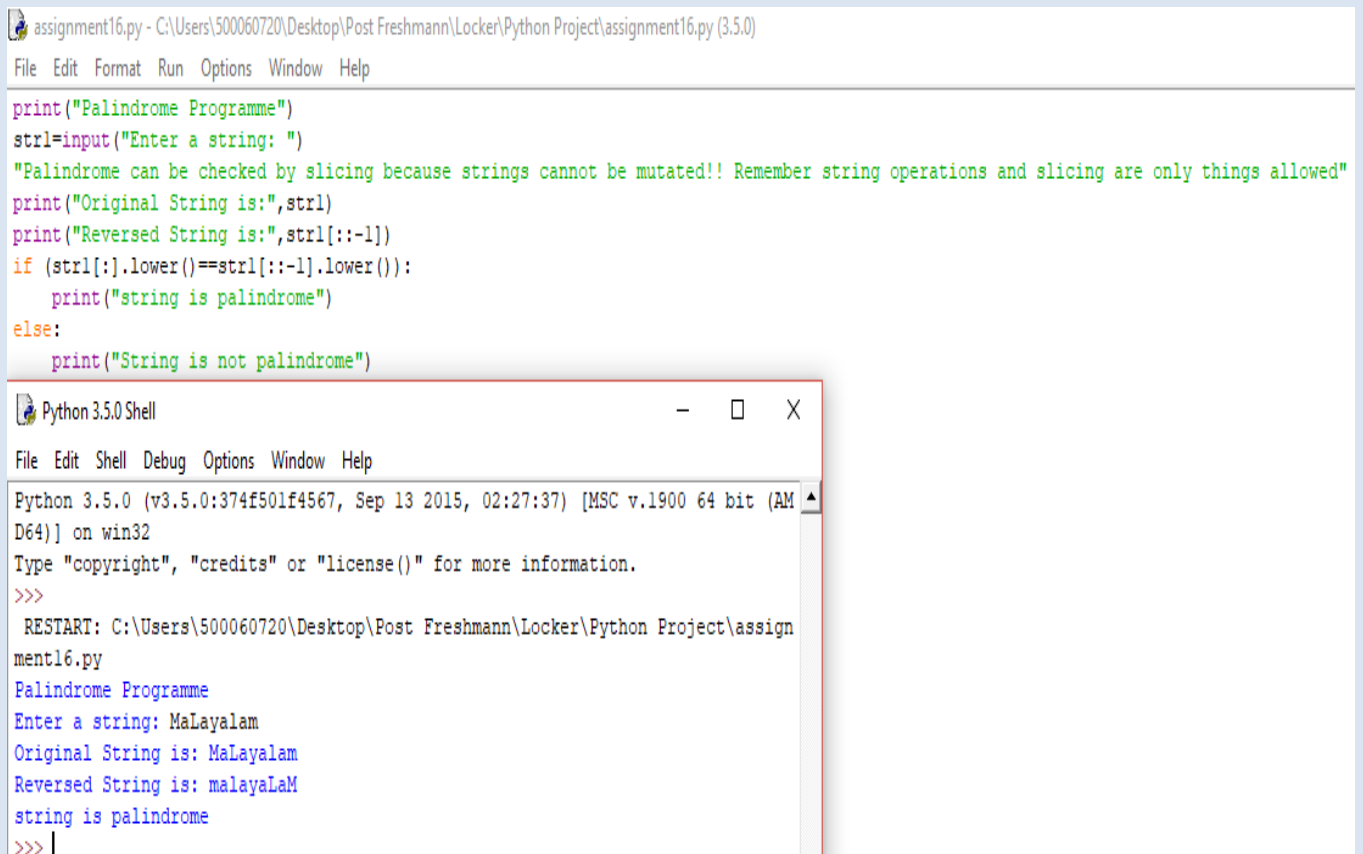
### Assignment 16

Duration : 10 mins

- Accept a string as an input from the user. Check if the accepted string is palindrome or not.
- If the string is palindrome, print "String is palindrome", otherwise print "String is not palindrome".
- Also print the actual and the reversed strings.

Note - Ignore the case of characters.

Hint - A palindrome string remains the same if the characters of the string are reversed.



```
assignment16.py - C:\Users\500060720\Desktop\Post Freshmann\Locker\Python Project\assignment16.py (3.5.0)
File Edit Format Run Options Window Help

print("Palindrome Programme")
str1=input("Enter a string: ")
"Palindrome can be checked by slicing because strings cannot be mutated!! Remember string operations and slicing are only things allowed"
print("Original String is:",str1)
print("Reversed String is:",str1[::-1])
if (str1[:].lower()==str1[::-1].lower()):
    print("string is palindrome")
else:
    print("String is not palindrome")

Python 3.5.0 Shell
File Edit Shell Debug Options Window Help
Python 3.5.0 (v3.5.0:374f501f4567, Sep 13 2015, 02:27:37) [MSC v.1900 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:\Users\500060720\Desktop\Post Freshmann\Locker\Python Project\assignment16.py
Palindrome Programme
Enter a string: MaLayalam
Original String is: MaLayalam
Reversed String is: maLaYaLaM
string is palindrome
>>>
```



## **Assignment 17**

### Assignment 17

Duration : 10 mins

Accept two strings 'string1' and 'string2' as an input from the user. Generate a resultant string, such that it is a concatenated string of all upper case alphabets from both the strings in the order they appear. Print the actual and the resultant strings.

Note: Each character should be checked if it is a upper case alphabet and then it should be concatenated to the resultant string.

Sample Input:

string1:	I Like C
string2:	Mary Likes Python
Output:	ILCMLP

assignment17.py - C:\Users\500060720\Desktop\Post Freshmann\Locker\Python Project\assignment17.py (3.5.0)

File Edit Format Run Options Window Help

```
print("Concatenation of capital letters of two strings in order they appear")
string1=input("Enter a string: ")
string2=input("Enter another string: ")
str1=str()
for x in string1:
    if x.isupper():
        str1=str1+x
for y in string2:
    if y.isupper():
        str1=str1+y
print(string1)
print(string2)
print(str1)
```

Python 3.5.0 Shell

File Edit Shell Debug Options Window Help

Python 3.5.0 (v3.5.0:374f501f4567, Sep 13 2015, 02:27:37) [MSC v.1900 64 bit  
D64)] on win32

Type "copyright", "credits" or "license()" for more information.

>>>

RESTART: C:\Users\500060720\Desktop\Post Freshmann\Locker\Python Project\ass  
ment17.py

Concatenation of capital letters of two strings in order they appear

Enter a string: Nishkarsh

Enter another string: Raj

Nishkarsh

Raj

NR

>>> |

## **Assignment 18**

### Assignment 18

Duration : 10 mins

Given a string containing both upper and lower case alphabets. Write a Python program to count the number of occurrences of each alphabet(case insensitive) and display the same.

Sample Input: ABaBCbGc

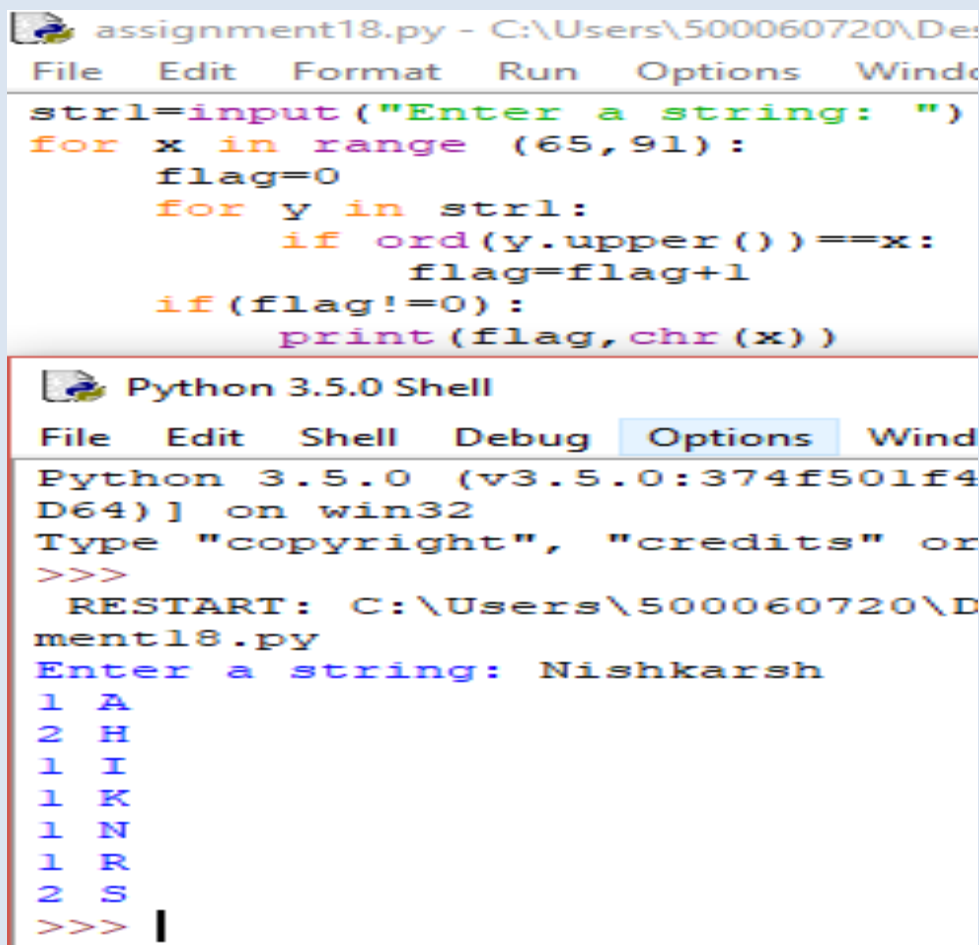
Sample Output:

2A

3B

2C

1G



```
assignment18.py - C:\Users\500060720\Desktop
File Edit Format Run Options Window
str1=input("Enter a string: ")
for x in range (65,91):
    flag=0
    for y in str1:
        if ord(y.upper())==x:
            flag=flag+1
    if(flag!=0):
        print(flag,chr(x))

Python 3.5.0 Shell
File Edit Shell Debug Options Window
Python 3.5.0 (v3.5.0:374f501f4D64) on win32
Type "copyright", "credits" or
>>>
RESTART: C:\Users\500060720\Desktop
ment18.py
Enter a string: Nishkarsh
1 A
2 H
1 I
1 K
1 N
1 R
2 S
>>> |
```

## **Assignment 19**

### Assignment 19

Duration : 10 mins

Write a Python program to accept a string 'accepted\_string'. Generate a resultant string 'resultant\_string' such that 'resultant\_string' should contain all characters at the even position of 'accepted\_string'(ignoring blank spaces). Display 'resultant\_string' in reverse order.

accepted\_string: An apple a day keeps the doctor away

resultant\_string: AapedyepeteotrwY

expected\_output: ywrtoetpeydepaA

Hint: String starts from index 0, hence the starting character is at even position.

assignment19.py - C:\Users\500060720\Desktop\Post Freshmann\Locker\F

File Edit Format Run Options Window Help

```
accepted_string=input("Enter a string: ")
str1=str() #empty string declaration
"Removing blank spaces in new string"
for x in accepted_string:
    if x==" ":
        continue
    else:
        str1=str1+x
resultant_string=str() #empty string declaration
"Adding even spaced values"
for x in range (0,len(str1)+1):
    if(x%2==0):
        resultant_string=resultant_string+str1[x]
    else:
        continue
print("accepted string: ",accepted_string)
print("resultant string: ",resultant_string)
print("expected string: ",resultant_string[::-1])
```

Python 3.5.0 Shell

File Edit Shell Debug Options Window Help

Python 3.5.0 (v3.5.0:374f501f4567, Sep 13 2015, 02:04:04) [ on win32

Type "copyright", "credits" or "license()" for more  
>>>

RESTART: C:\Users\500060720\Desktop\Post Freshmann\Locker\Freshmann\assignment19.py

```
Enter a string: Nishkarsh Raj Khare
accepted string:  Nishkarsh Raj Khare
resultant string:  NskrhaKae
expected string:  eaKahrksN
>>> |
```

## **Assignment 20**

### Assignment 20

Duration : 5 mins

Write a Python program to generate first 'n' Fibonacci numbers where 'n' is accepted as an input from the user.

Store the generated Fibonacci numbers in a list and display the output.

Sample input: 5

Sample output: [0, 1, 1, 2, 3]

assignment20.py - C:\Users\500060720\Desktop\Post Freshmann\Locker\Python Project\assignment20.py (3.5.0)

File Edit Format Run Options Window Help

```
print("Fibonacci code for lists")
a,b=0,1
z=0
fibb_list=[]
fibb_list.insert(0,a)
fibb_list.insert(1,b)
z=z+1
num=int(input("Enter number of terms in Fibonacci series you want to store: "))
for i in range(0,num-1):
    z=z+1
    fibb_list.insert(z,a+b)
    a=b
    b=fibb_list[z]
for i in range (0,num):
    print(fibb_list[i],end=" ")
```

Python 3.5.0 Shell

File Edit Shell Debug Options Window Help

Python 3.5.0 (v3.5.0:374f501f4567, Sep 13 2015, 02:27:37) [MSC v.1900 64 bit (AMD64)] on win32

Type "copyright", "credits" or "license()" for more information.

>>>

RESTART: C:\Users\500060720\Desktop\Post Freshmann\Locker\Python Project\assignment20.py

Fibonacci code for lists

Enter number of terms in Fibonacci series you want to store: 6

0 1 1 2 3 5

>>> |

## **Assignment 21**

### Assignment 21

Duration : 15 mins

The "Variety Retail Store" sells different varieties of Furniture to the customers. The list of furniture available with its respective cost is given below:

Furniture	Sofa set	Dining table	T.V. Stand	Cupboard
Cost in Rs.	20,000	8,500	4,599	13,920

The furniture and its corresponding cost should be stored as a list. A customer can order any furniture in any quantity (the name and quantity of the furniture will be provided). If the required furniture is available in the furniture list(given above) and quantity to be purchased is greater than zero, then bill amount should be calculated. In case of invalid values for furniture required by the customer and quantity to be purchased, display appropriate error message and consider bill amount to be 0. Initialize required furniture and quantity with different values and test the results.

Write a Python program to calculate and display the bill amount to be paid by the customer based on the furniture bought and quantity purchased.

Hint - Create two different lists for 'Furniture' and 'Cost'. Indices of two lists should be matched to retrieve the cost of a particular furniture.

assignment21.py - C:\Users\500060720\Desktop\Post Freshmann\Locker\Python Project\ass

File Edit Format Run Options Window Help

```
print("Variety Retail Store")
furniture=['Sofa Set','Dining Table','T.V. Stand','Cupboard']
cost_in_rupees=[20000,8500,4599,13920]
name=input("Enter name of the item you want to buy: ")
amount=int(input("Enter the quantity of the item: "))
flag=0
if amount<0:
    print("Amount cannot be negative!!!")
    bill=0
else:
    for x in range(0,3):
        if furniture[x]==name:
            z=x
            flag=1
    if flag==0:
        print("Item not in the inventory")
        bill=0
    else:
        bill=cost_in_rupees[z]*amount
print("The total bill is:",bill)
```

Python 3.5.0 Shell

File Edit Shell Debug Options Window Help

Python 3.5.0 (v3.5.0:374f501f4567, Sep 13 2015, 02:27:37) [MSC  
D64] on win32

Type "copyright", "credits" or "license()" for more informatio  
>>>

RESTART: C:\Users\500060720\Desktop\Post Freshmann\Locker\Pyt  
ment21.py

Variety Retail Store

Enter name of the item you want to buy: Dining Table

Enter the quantity of the item: 24

The total bill is: 204000

>>> |



## **Assignment 22**

### Assignment 22

Duration : 10 mins

Consider the list of courses opted by a Student "John" and available electives at ABC Training Institute:

```
courses = ("Python Programming", "RDBMS", "Web Technology", "Software Engg.")
```

```
electives = ("Business Intelligence", "Big Data Analytics")
```

Write a Python Program to satisfy business requirements mentioned below:

1. List the number of courses opted by John.
2. List all the courses opted by John.
3. John is also interested in elective courses mentioned above. Print the updated tuple including electives.

assignment22.py - C:\Users\500060720\Desktop\Post Freshmann\Locker\Python Project\assignment22.py (3.5.0)

File Edit Format Run Options Window Help

```
name="John"
institute="ABC Training Institute"
courses=("Python Programming","RDBMS","Web Technology","Software Engg.")
electives=("Business Intelligence","Big Data Analytics")
print(("Number of courses opted by %s are %d " % (name, len(courses))))
print(("Courses opted by %s are" % (name)))
for i in range(0,4):
    print(courses[i],end=" ")
'''courses.insert(5,electives[0])
courses.insert(6,electives[1])'''
print("")

'''Are you kidding me!!!! Tuples are immutable! First typecast as list change and then remake as tuple'''

courses=list(courses)
courses.append(electives[0])
courses.append(electives[1])
courses=tuple(courses)

print("Added elective with complete list are now shown")
for i in range(0,6):
    print(courses[i],end=" ")
```

Python 3.5.0 Shell

File Edit Shell Debug Options Window Help

Python 3.5.0 (v3.5.0:374f501f4567, Sep 13 2015, 02:27:37) [MSC v.1900 64 bit (AMD64)] on win32

Type "copyright", "credits" or "license()" for more information.

>>>

RESTART: C:\Users\500060720\Desktop\Post Freshmann\Locker\Python Project\assignment22.py

Number of courses opted by John are 4

Courses opted by John are

Python Programming RDBMS Web Technology Software Engg.

Added elective with complete list are now shown

Python Programming RDBMS Web Technology Software Engg. Business Intelligence Big Data Analytics

>>> |

## **Assignment 23**

### Assignment 23

Duration : 10 mins

Given below is a dictionary 'customer\_details' representing customer details from a Retail Application. Customer Id is the key and Customer Name is the value.

```
customer_details = { 1001 : "John", 1004 : "Jill", 1005: "Joe", 1003 : "Jack" }
```

Write Python code to perform the operations mentioned below:

- a) Print details of customers.
- b) Print number of customers.
- c) Print customer names in ascending order.
- d) Delete the details of customer with customer id = 1005 and print updated dictionary.
- e) Update the name of customer with customer id = 1003 to "Mary" and print updated dictionary.
- f) Check whether details of customer with customer id = 1002 exists in the dictionary.

assignment23.py - C:\Users\500060720\Desktop\Post Freshmann\Locker\Python Project\assignment23.py (3.5.0)

File Edit Format Run Options Window Help

```
customer_details = {1001:"John",1004:"Jill",1005:"Joe",1003:"Jack"}
flag=0
print("a) Customer Details")
for k in customer_details:
    print("Name of the ",k,"Id customer is",customer_details[k])
print("b) Number of customers")
print("Number of customers are:",len(customer_details))
print("c) Print customer names in ascending order")
import operator
sorted_cust_name=sorted(customer_details.items(),key=operator.itemgetter(1)) #sorted_cust_name is a tuple
for k in range(0,3):
    print(sorted_cust_name[k],end=" ")
print("")
print("d) Delete the details of customer with customer id=1005 and print updated dictionary")
del customer_details[1005]
for k in customer_details:
    print("Name of the ",k,"Id customer is",customer_details[k])
print('e) Update the name of customer with customer id = 1003 to "Mary" and print updated dictionary.')
customer_details[1003]="Mary"
for k in customer_details:
    print("Name of the ",k,"Id customer is",customer_details[k])
print("f) Check whether details of customer with customer id = 1002 exists in the dictionary.")
for k in customer_details:
    if k==1002:
        flag=1
        break
if flag==1:
    print("1002 element record exists in the dictionary")
else:
    print("Record not found")
```

Python 3.5.0 Shell

File Edit Shell Debug Options Window Help

Python 3.5.0 (v3.5.0:374f501f4567, Sep 13 2015, 02:27:37) [MSC v.1900 64 bit (AMD64)] on win32  
Type "copyright", "credits" or "license()" for more information.

>>>

RESTART: C:\Users\500060720\Desktop\Post Freshmann\Locker\Python Project\assignment23.py

a) Customer Details

Name of the 1001 Id customer is John

Name of the 1003 Id customer is Jack

Name of the 1004 Id customer is Jill

Name of the 1005 Id customer is Joe

b) Number of customers

Number of customers are: 4

c) Print customer names in ascending order

(1003, 'Jack') (1004, 'Jill') (1005, 'Joe')

d) Delete the details of customer with customer id=1005 and print updated dictionary

Name of the 1001 Id customer is John

Name of the 1003 Id customer is Jack

Name of the 1004 Id customer is Jill

e) Update the name of customer with customer id = 1003 to "Mary" and print updated dictionary.

Name of the 1001 Id customer is John

Name of the 1003 Id customer is Mary

Name of the 1004 Id customer is Jill

f) Check whether details of customer with customer id = 1002 exists in the dictionary.

Record not found

>>> |

## **Assignment 24**

### Assignment 24

Duration : 10 mins

Consider a scenario from ABC Training Institute. The given table shows the marks scored by students of grade XI in Python Programming course.

Student Name	Marks Scored
John	86.5
Jack	91.2
Jill	84.5
Harry	72.1
Joe	80.5

Write a Python program to meet the requirements mentioned below:

a. Display the name and marks for every student.

assignment24.py - C:\Users\500060720\Desktop\Post Freshmann\Locker\Python Project\assignment24.py (3.5.0)

File Edit Format Run Options Window Help

```
print("Python Programming Course Marksheet")
marks={"John":86.5,"Jack":91.2,"Jill":84.5,"Harry":72.1,"Joe":80.5}
for k in marks:
    print(("Name of the student is %s and has scored %0.2f marks"%(k,marks[k]))
import operator
sorted_marks = sorted (marks.items(),key=operator.itemgetter(1))
for k in range(0,2):
    print("Highest Marks at",k+1,"position are",sorted_marks[len(marks)-1-k])
sum=0
for k in marks:
    sum=sum+marks[k]
average=sum/len(marks)
print("Average score of the class is",average)
```

Python 3.5.0 Shell

File Edit Shell Debug Options Window Help

Python 3.5.0 (v3.5.0:374f501f4567, Sep 13 2015, 02:27:37) [MSC v.1900 64 bit (AMD64)] on win32

Type "copyright", "credits" or "license()" for more information.

>>>

RESTART: C:\Users\500060720\Desktop\Post Freshmann\Locker\Python Project\assignment24.py

Python Programming Course Marksheet

Name of the student is John and has scored 86.50 marks

Name of the student is Joe and has scored 80.50 marks

Name of the student is Jill and has scored 84.50 marks

Name of the student is Jack and has scored 91.20 marks

Name of the student is Harry and has scored 72.10 marks

Highest Marks at 1 position are ('Jack', 91.2)

Highest Marks at 2 position are ('John', 86.5)

Average score of the class is 82.96

>>> |

## **Assignment 25**

### Assignment 25

Duration : 20 mins

Consider the scenario from "Variety Retail Store" discussed in 'List' section. The list of furniture available with its respective cost is given below:

Furniture	Sofa set	Dining table	T.V. Stand	Cupboard
Cost in Rs.	20,000	8,500	4,599	13,920

A customer can order any furniture in any quantity. If the required furniture is available in the furniture list(given above) and quantity to be purchased is greater than zero, then bill amount should be calculated. In case of invalid values for furniture required by the customer and quantity to be purchased, display appropriate error message and consider bill amount to be 0. Initialize required furniture and quantity with different values and test the results.

Calculate and display the bill amount to be paid by the customer based on the furniture bought and quantity



assignment25.py - C:\Users\500060720\Desktop\Post Freshmann\Locker\Python Project\assignment25.py (3.5.0)

File Edit Format Run Options Window Help

```
print("Variety Retail Store")
dict={"Sofa Set":20000,"Dining Table":8500,"T.V. Stand":4599,"Cupboard":13920}
flag=0
for k in dict:
    print("The items available are")
    print("Item:",k,"is available with base price:",dict[k])
name=input("Enter the name of the item you want to buy: ")
qty=int(input("Enter the quantity of item you want to buy: "))
if qty<0:
    print("Quantity cannot be less than zero")
    bill=0
else:
    for k in dict:
        if k==name:
            x=dict[k]
            flag=1
            break
    if flag==0:
        print("Item is not in the list")
        bill=0
    else:
        bill=dict[k]*qty
print("Total amount to be paid is",bill)
```

Python 3.5.0 Shell

File Edit Shell Debug Options Window Help

Python 3.5.0 (v3.5.0:374f501f4567, Sep 13 2015, 02:27:37)  
Type "copyright", "credits" or "license()" for more inform  
>>>  
RESTART: C:\Users\500060720\Desktop\Post Freshmann\Locker  
Variety Retail Store  
The items available are  
Item: T.V. Stand is available with base price: 4599  
The items available are  
Item: Cupboard is available with base price: 13920  
The items available are  
Item: Sofa Set is available with base price: 20000  
The items available are  
Item: Dining Table is available with base price: 8500  
Enter the name of the item you want to buy: Dining Table  
Enter the quantity of item you want to buy: 52  
Total amount to be paid is 442000  
>>> |

assignment25\_1.py - C:\Users\500060720\Desktop\Post Freshmann\Locker\Python Project\assignment25\_1.py (3.5.0)

File Edit Format Run Options Window Help

```
print("Variety Retail Store")
flag=0
list= [("Sofa set",20000), ("Dining Table",8500), ("T.V. Stand",4599), ("Cupboard",13920)]
print("List of items available are")
for k in range(0,len(list)):
    print(list[k])
name=input("Enter the name of the item you want to buy: ")
qty=int(input("Enter the amount of items you want to buy: "))
if qty<0:
    print("Quantity cannot be less than zero")
    bill=0
else:
    for k in range(0,len(list)):
        if list[k][0]==name:
            x=list[k][1]
            flag=1
            break
    if flag==0:
        print("Item not available")
        bill=0
    else:
        bill=qty*x
print("Total amount to be paid is",bill)
```

Python 3.5.0 Shell

File Edit Shell Debug Options Window Help

Python 3.5.0 (v3.5.0:374f501f4567, Sep 13 2015, 02:27:37)

Type "copyright", "credits" or "license()" for more infor

>>>

RESTART: C:\Users\500060720\Desktop\Post Freshmann\Locker

Variety Retail Store

List of items available are

('Sofa set', 20000)

('Dining Table', 8500)

('T.V. Stand', 4599)

('Cupboard', 13920)

Enter the name of the item you want to buy: Dining Table

Enter the amount of items you want to buy: 52

Total amount to be paid is 442000

>>> |

## **Assignment 26**

### Assignment 26

Duration :

Consider a scenario from ABC Training Institute. Given below are two Sets representing the names of students enrolled for a particular course:

```
java_course = {"John", "Jack", "Jill", "Joe"}
```

```
python_course = {"Jake", "John", "Eric", "Jill"}
```

Write a Python program to list the number of students enrolled for:

- 1) Python course
- 2) Java course only
- 3) Python course only
- 4) Both Java and Python courses
- 5) Either Java or Python courses but not both
- 6) Either Java or Python courses

assignment26.py - C:\Users\500060720\Desktop\Post Freshmann\Locker\Python Project\assignment26.py (3.5.0)

File Edit Format Run Options Window Help

```
institute="ABC Training Institue"
java_course={"John","Jack","Jill","Joe"}
python_course={"Jake","John","Eric","Jill"}
print("Number of students enrolled in python course are",len(python_course))
print("Number of students enrolled in Java course only are",len(java_course-python_course))
print("Number of students enrolled in Python course only are",len(python_course-java_course))
print("Number of students enrolled in both Java and Python course are",len(java_course&python_course))
print("Number of students enrolled in either Java or Python but not in both course are",len((python_course|java_course)-python_course&java_course))
print("Number of students enrolled in either Java or Python courses are",len(python_course&java_course))
```

Python 3.5.0 Shell

- □ ×

File Edit Shell Debug Options Window Help

Python 3.5.0 (v3.5.0:374f501f4567, Sep 13 2015, 02:27:37) [MSC v.1900 64 bit (AMD64)]

on win32

Type "copyright", "credits" or "license()" for more information.

>>>

RESTART: C:\Users\500060720\Desktop\Post Freshmann\Locker\Python Project\assignment26.py

Number of students enrolled in python course are 4

Number of students enrolled in Java course only are 2

Number of students enrolled in Python course only are 2

Number of students enrolled in both Java and Python course are 2

Number of students enrolled in either Java or Python but not in both course are 2

Number of students enrolled in either Java or Python courses are 2

>>> |

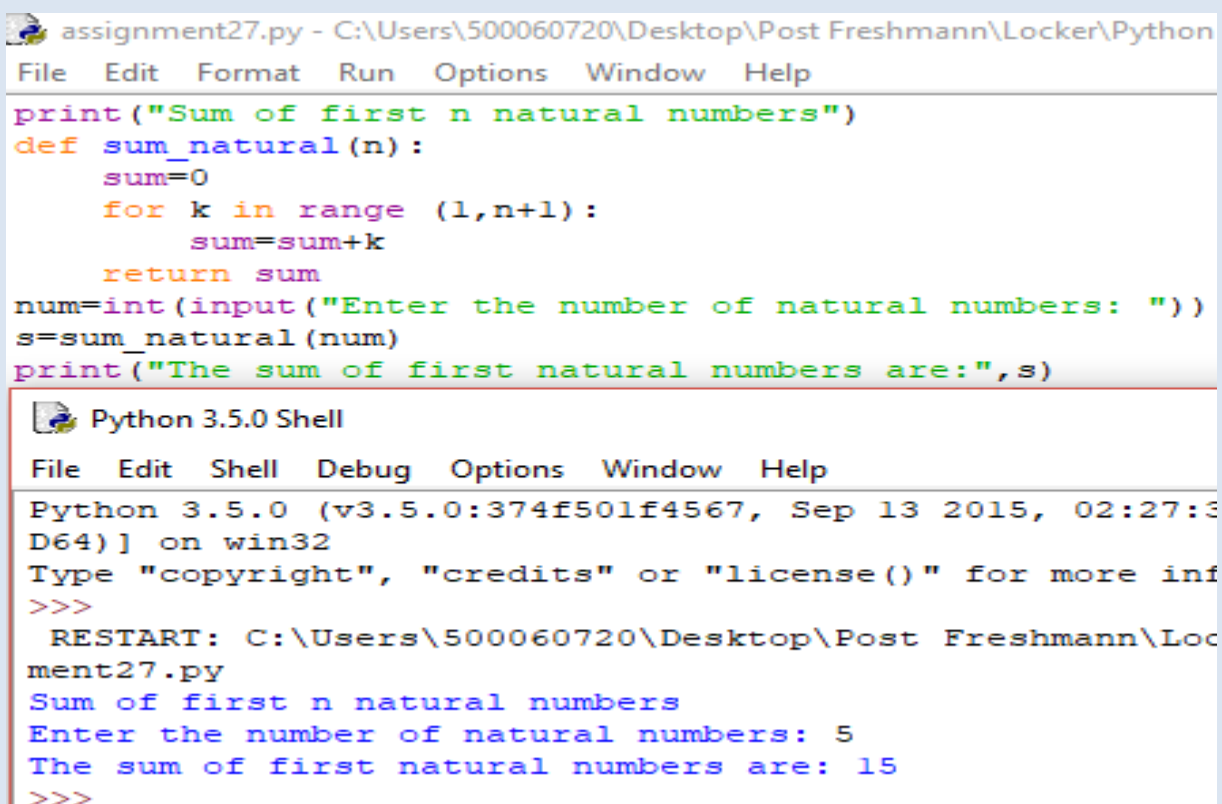
## **Assignment 27**

### Assignment 27

Using functions, re-write and execute Python program to:

1. Add natural numbers upto n where n is taken as an input from user.
2. Print Fibonacci series till nth term (Take input from user).

Note: You have implemented these programs using loops earlier.



```
assignment27.py - C:\Users\500060720\Desktop\Post Freshmann\Locker\Python
File Edit Format Run Options Window Help
print("Sum of first n natural numbers")
def sum_natural(n):
    sum=0
    for k in range (1,n+1):
        sum=sum+k
    return sum
num=int(input("Enter the number of natural numbers: "))
s=sum_natural(num)
print("The sum of first natural numbers are:",s)

Python 3.5.0 Shell
File Edit Shell Debug Options Window Help
Python 3.5.0 (v3.5.0:374f501f4567, Sep 13 2015, 02:27:3
D64)] on win32
Type "copyright", "credits" or "license()" for more inf
>>>
RESTART: C:\Users\500060720\Desktop\Post Freshmann\Loc
ment27.py
Sum of first n natural numbers
Enter the number of natural numbers: 5
The sum of first natural numbers are: 15
>>>
```

assignment27\_1.py - C:\Users\500060720\Desktop\Post Freshmann\Locker\Python Project\assignment27\_1.py (3.5.0)

File Edit Format Run Options Window Help

```
print("Printing Fibonacci series till nth term")
def print_fibonacci(num):
    a,b=0,1
    print(a,end=" ")
    print(b,end=" ")
    for k in range(1,num-1):
        c=a+b
        print(c,end=" ")
        a=b
        b=c
    print("")
    return
n=int(input("Enter a number of terms until which you want to generate Fibonacci Series: "))
print_fibonacci(n)
```

Python 3.5.0 Shell

File Edit Shell Debug Options Window Help

Python 3.5.0 (v3.5.0:374f501f4567, Sep 13 2015, 02:27:37) [MSC v.1900 64 bit (AMD64)] on win32

Type "copyright", "credits" or "license()" for more information.

>>>

RESTART: C:\Users\500060720\Desktop\Post Freshmann\Locker\Python Project\assignment27\_1.py

Printing Fibonacci series till nth term

Enter a number of terms until which you want to generate Fibonacci Series: 6

0 1 1 2 3 5

>>>

## **Assignment 28**

### Assignment 28

At an airport, a traveler is allowed entry into the flight only if he clears the following checks:

1. Baggage Check
2. Immigration Check
3. Security Check

The logic for the check methods are given below:

#### **check\_baggage (baggage\_weight)**

- returns True if *baggage\_weight* is greater than or equal to 0 and less than or equal to 40. Otherwise returns False.

#### **check\_immigration (expiry\_year)**

- returns True if *expiry\_year* is greater than or equal to 2001 and less than or equal to 2025. Otherwise returns False.

#### **check\_security(noc\_status)**

- returns True if *noc\_status* is 'valid' or 'VALID', for all other values return False.

#### **traveler()**

- Initialize the traveler Id and traveler name and invoke the functions `check_baggage()`, `check_immigration()` and `check_security()` by passing required arguments.
- Refer the table below for values of arguments.
- If all values of `check_baggage()`, `check_immigration()` and `check_security()` are true,

display traveler\_id and traveler\_name

display "Allow Traveler to fly!"

Otherwise,

display traveler\_id and traveler\_name

display "Detain Traveler for Re-checking!"

Variable	Value
traveler_id	1001
traveler_name	Jim
baggage_weight	35
expiry_year	2019
noc_status	VALID


Invoke the `traveler()` function. Modify the values of different variables in `traveler()` function and observe the output.

assignment28.py - C:\Users\500060720\Desktop\Post Freshmann\Locker\Python Project\assignment28.py (3.5.0)

File Edit Format Run Options Window Help

```
def check_baggage(baggage_weight):
    if baggage_weight>=0 and baggage_weight<=40:
        return True
    else:
        return False
def check_immigration(expiry_year):
    if expiry_year>=2001 and expiry_year<=2025:
        return True
    else:
        return False
def check_security(noc_status):
    if noc_status=="valid" or noc_status=="VALID":
        return True
    else:
        return False
def traveler():
    traveler_id=input("Enter the Traveler ID: ")
    traveler_name=input("Enter the Traveler name: ")
    baggage_weight=int(input("Enter the baggage weight: "))
    expiry_year=int(input("Enter the expiry year of immigration: "))
    noc_status=input("Enter the noc status of the traveler: ")
    print(traveler_id)
    print(traveler_name)
    if check_baggage(baggage_weight)==True and check_immigration(expiry_year)==True and check_security(noc_status)==True:
        print("Allow traveler to fly!")
    else:
        print("Detain traveler for re-checking")
traveler()
```



 Python 3.5.0 Shell

File Edit Shell Debug Options Window Help

Python 3.5.0 (v3.5.0:374f501f4567, Sep 13 2015)

Type "copyright", "credits" or "license()" for more

>>>

RESTART: C:\Users\500060720\Desktop\Post Free

Enter the Traveler ID: 500060720

Enter the Traveler name: Nishkarsh Raj

Enter the baggage weight: 10

Enter the expiry year of immigration: 2020

Enter the noc status of the traveler: valid

500060720

Nishkarsh Raj

Allow traveler to fly!

>>> |

## **Assignment 29**

### Assignment 29

Duration : 10 mins

Consider the pseudo code for generating Fibonacci series using Recursion:

FIBO (number)

1. if (number = 0) then
2.     return (0)
3. else if (number = 1) then
4.     return (1)
5. else
6.     return FIBO(number - 1) + FIBO(number - 2)
7. end if

Write a program in Python to implement the same using Recursion and execute it in Eclipse. Print appropriate error message if the user enters negative number as input.

```
assignment29.py - C:\Users\500060720\Desktop\Post Freshmann\Locker\Python Project\assignment29.py (3.5.0)
File Edit Format Run Options Window Help

print("Fibonacci Numbers using Recursion")
def FIBO(number):
    sum=0
    if number==0:
        return 0
    elif number==1:
        return 1
    else:
        sum+= FIBO(number-1)+FIBO(number-2)
    return sum
num=int(input("Enter a number to find fibonacci series until given number: "))
if num<0:
    print("Negative number not allowed")
else:
    s=FIBO(num)
    print(s)

Python 3.5.0 Shell
File Edit Shell Debug Options Window Help
Python 3.5.0 (v3.5.0:374f501f4567, Sep 13 2015, 02:27:37) [MSC v.1900 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:\Users\500060720\Desktop\Post Freshmann\Locker\Python Project\assignment29.py
Fibonacci Numbers using Recursion
Enter a number to find fibonacci series until given number: 6
8
>>>
```

## **Assignment 30**

### Assignment 30

Duration : 30 mins

Write a Python program to implement the following (Use Recursion):

1. Print first 'n' multiples of 3, where 'n' is taken as an input from the user. The multiples should be printed from first to last.
2. Reverse a string. Print the original and reversed string.
3. Check if the given string is palindrome. If yes, print "String is palindrome" otherwise print "String is not palindrome".

assignment30\_1.py - C:\Users\500060720\Desktop\Post Freshmann\Locker\Python Project\assign

File Edit Format Run Options Window Help

```
def multiple_3(num,z=1):  
    if num==0:  
        return  
    else:  
        print("The",z,"multiple of 3 is:",3*z)  
        z=z+1  
        multiple_3(num-1,z)  
n=int(input("Enter number until which you want multiple of 3: "))  
multiple_3(n)
```

Python 3.5.0 Shell

File Edit Shell Debug Options Window Help

Python 3.5.0 (v3.5.0:374f501f4567, Sep 13 2015, 02:27:37) [MSC v.1  
D64] on win32

Type "copyright", "credits" or "license()" for more information.

>>>

RESTART: C:\Users\500060720\Desktop\Post Freshmann\Locker\Python  
ment30\_1.py

Enter number until which you want multiple of 3: 15

The 1 multiple of 3 is: 3

The 2 multiple of 3 is: 6

The 3 multiple of 3 is: 9

The 4 multiple of 3 is: 12

The 5 multiple of 3 is: 15

The 6 multiple of 3 is: 18

The 7 multiple of 3 is: 21

The 8 multiple of 3 is: 24

The 9 multiple of 3 is: 27

The 10 multiple of 3 is: 30

The 11 multiple of 3 is: 33

The 12 multiple of 3 is: 36

The 13 multiple of 3 is: 39

The 14 multiple of 3 is: 42

The 15 multiple of 3 is: 45

>>> |

```
assignment30_2.py - C:\Users\500060720\Desktop\Post Freshma
File Edit Format Run Options Window Help
print("Reversing a string using recursion")
def reverse(s):
    if s=="":
        return s
    else:
        return reverse(s[1:])+s[0]
str=input("Enter a string: ")
rstr=reverse(str)
print("The reversed string is:",rstr)

Python 3.5.0 Shell
File Edit Shell Debug Options Window Help
Python 3.5.0 (v3.5.0:374f501f4567, Sep 13 20
D64)] on win32
Type "copyright", "credits" or "license()" f
>>>
RESTART: C:\Users\500060720\Desktop\Post Fr
ment30_2.py
Reversing a string using recursion
Enter a string: Nishkarsh Raj Khare
The reversed string is: erahK jaR hsrakhsiN
>>> |
```

```
assignment30_3.py - C:\Users\500060720\Desktop\Post Freshmar
File Edit Format Run Options Window Help
print("Reversing a string using recursion")
def reverse(s):
    if s=="":
        return s
    else:
        return reverse(s[1:])+s[0]
str=input("Enter a string: ")
rstr=reverse(str)
print("The reversed string is:",rstr)
if str==rstr:
    print("The string is palindrome")
else:
    print("The string is not palindrome")

Python 3.5.0 Shell
File Edit Shell Debug Options Window Help
Python 3.5.0 (v3.5.0:374f501f4567, Sep 13 20
D64)] on win32
Type "copyright", "credits" or "license()" f
>>>
RESTART: C:\Users\500060720\Desktop\Post Fr
ment30_3.py
Reversing a string using recursion
Enter a string: Nishkarsh Raj khare
The reversed string is: erahk jaR hsrakhsiN
The string is not palindrome
>>>
```

## **Assignment 31**

### Assignment 31

Write a Python program to:

1. read a file.
2. add backslash (\) before every double quote in the file contents.
3. write it to another file in the same folder.
4. print the contents of both the files.

For example:

If the first file is 'TestFile1.txt' with text as:

Jack said, "Hello Pune".

The output of the file 'TestFile2.txt' should be:

Jack said,\"Hello Pune\".

assignment31.py - C:\Users\500060720\Desktop\Post Freshmann\Locker\Python Project\assignment3

File Edit Format Run Options Window Help

```
file=open("testfile1.txt","r",1)
print("1)Read a file")
s=file.read()
print(s)
print("")
print("")
print("2) Add backslash\ before every double quote in file contents")
list1=s.split('"')
str1=str()
for k in range (0,len(list1)):
    if k!=len(list1)-1:
        str1=str1+list1[k]+'\\'
    else:
        str1=str1+list1[k]
print(str1)
print("")
print("")
print("3) Write it to another file in the same folder")
file.close()
file=open("testfile2.txt","w",1)
file.write(str1)
print("")
print("")
print("4) Write contents of both the files")
print(str1)
print(s)
```



## Python 3.5.0 Shell

File Edit Shell Debug Options Window Help

Python 3.5.0 (v3.5.0:374f501f4567, Sep 13 2015, 02:27:37) [MS  
Type "copyright", "credits" or "license()" for more informati  
>>>

RESTART: C:\Users\500060720\Desktop\Post Freshmann\Locker\Py  
1) Read a file  
Jack said, "Hello Pune"

2) Add backslash\ before every double quote in file contents  
Jack said, \"Hello Pune\"

3) Write it to another file in the same folder

4) Write contents of both the files

Jack said, \"Hello Pune\"

Jack said, "Hello Pune"

>>> |

## **Assignment 32**


### Assignment 32

Consider a file 'courses.txt' in D Drive with the following details:



```
Java  
Python  
Javascript  
PHP
```

Write a program to read the file and store the courses in Python variables as a:


- 1) Dictionary ( Sample - {0: 'Java', 1: 'Python', 2:'Javascript' 3: 'PHP'} )
- 2) List ( Sample - ['Java', 'Python', 'Javascript', 'PHP'] )

 assignment32\_1.py - C:\Users\500060720\Desktop\Post Freshmann\Locker\Python  
File Edit Format Run Options Window Help  

```
print("Storing text file as dictionary")
file=open("courses.txt","r+")
lines=file.readlines() #takes line by line with linefeed
dict={}
i=0
for line in lines:
    dict[i]=line.strip() #strip removes linefeed
    i=i+1
print(dict)
file.close()
```

  
 Python 3.5.0 Shell  
File Edit Shell Debug Options Window Help  
Python 3.5.0 (v3.5.0:374f501f4567, Sep 13 2015, 02:27:37) [MSC v.1900 64 bit (AMD64)] on win32  
Type "copyright", "credits" or "license()" for more information.  
>>>  
RESTART: C:\Users\500060720\Desktop\Post Freshmann\Locker\Python Project\assignment32\_1.py  
Storing text file as dictionary  
{0: 'Java', 1: 'Python', 2: 'Javascript', 3: 'PHP'}  
>>> assignment32\_2.py - C:\Users\500060720\Desktop\Post Freshmann\Locker\Python Project\assignment32\_2.py (3.5.0)  
File Edit Format Run Options Window Help  

```
print("Storing text file as list")
file=open("courses.txt","r+")
lines=file.readlines() #takes line by line with linefeed
list=[None]*len(lines) #what does this line means! why not simple empty list
i=0
for line in lines:
    list[i]=line.strip() #strip removes linefeed
    i=i+1
print(list)
file.close()
```

  
 Python 3.5.0 Shell  
File Edit Shell Debug Options Window Help  
Python 3.5.0 (v3.5.0:374f501f4567, Sep 13 2015, 02:27:37) [MSC v.1900 64 bit (AMD64)] on win32  
Type "copyright", "credits" or "license()" for more information.  
>>>  
RESTART: C:\Users\500060720\Desktop\Post Freshmann\Locker\Python Project\assignment32\_2.py  
Storing text file as list  
['Java', 'Python', 'Javascript', 'PHP']  
>>>

## **Assignment 33**

### Assignment 33

Duration : 20 mins

Consider a file 'student\_details.txt' in D Drive with the details of students in ABC institute - student id and name:

```
101 Rahul  
102 Julie  
103 Helena  
104 Kally
```

Write a program to read the file and store the student records in Python variable as:

- 1) List of lists
- 2) List of dictionaries

assignment33.py - C:\Users\500060720\Desktop\Post Freshmann\Locker\Python Project\assignment33.py (3.5.0)

File Edit Format Run Options Window Help

```
print("Storing text file as list of list")
file=open("assignment33.txt","r+")
lines=file.readlines() #takes line by line with linefeed
list1=[None]*len(lines) #what does this line means! why not simple empty list
i=0
for line in lines:
    list1[i]=line.strip() #strip removes linefeed
    i=i+1
for k in range(0,len(list1)):
    list1[k]=list1[k].split()
print(list1)
file.close()
```

Python 3.5.0 Shell

File Edit Shell Debug Options Window Help

Python 3.5.0 (v3.5.0:374f501f4567, Sep 13 2015, 02:27:37) [MSC v.1900 64 bit  
D64] on win32

Type "copyright", "credits" or "license()" for more information.

>>>

RESTART: C:\Users\500060720\Desktop\Post Freshmann\Locker\Python Project\ass  
ment33.py

Storing text file as list of list

[['101', 'Rahul'], ['102', 'Julie'], ['103', 'Helena'], ['104', 'Kally']]

>>> |

## **Assignment 34**

### Assignment 34

Duration : 20 mins

Consider a file 'rhyme.txt' in D Drive with following text:

```
Jingle bells jingle bells
Jingle all the way
Oh what fun it is to ride
In a one horse open sleigh
Jingle bells jingle bells
Jingle all the way
```

Write a Python program to count the words in the file using a dictionary (use space as a delimiter). Find unique words and the count of their occurrences(ignoring case). Write the output in another file "words.txt" at the same location.

Python 3.5.0 Shell

File Edit Shell Debug Options Window Help

Python 3.5.0 (v3.5.0:374f501f4567, Sep 13 2015, 02:27:37) [MSC v.1900 64 bit (AMD64)] on win32

Type "copyright", "credits" or "license()" for more information.

>>>

RESTART: C:\Users\500060720\Desktop\Post Freshmann\Locker\Python Project\assignment34.py

```
{0: 'Jingle bells jingle bells', 1: 'Jingle all the way', 2: 'Oh what fun it is to ride', 3: 'In a one horse open sleigh', 4: 'Jingle bells jingle bells', 5: 'Jingle all the way'}
```

Number of words in the file are: 29

Unique occurrences are:

```
{'it': 1, 'way': 2, 'jingle': 6, 'to': 1, 'one': 1, 'a': 1, 'in': 1, 'oh': 1, 'fun': 1, 'is': 1, 'bells': 4, 'all': 2, 'horse': 1, 'what': 1, 'sleigh': 1, 'open': 1, 'the': 2, 'ride': 1}
```

>>> |

assignment34.py - C:\Users\500060720\Desktop\Post Freshmann\Locker\Python Project\assignment34.py (3.5.0)

File Edit Format Run Options Window Help

```
file=open("rhyme.txt","r+")
dictl={}
i=0
lines=file.readlines()
for line in lines:
    dictl[i]=line.strip()
    i=i+1
print(dictl)
l=0
strl=str()
for k in dictl:
    strl=strl+dictl[k]+" "
strl=strl.lower()
#print(strl)
for k in dictl:
    dictl[k]=dictl[k].split()
    l=l+len(dictl[k])
print("Number of words in the file are:",l)
def word_count(str):
    counts=dict()
    words=str.split()
    for word in words:
        if word in counts:
            counts[word]+=1
        else:
            counts[word]=1
    return counts
print("Unique occurences are:")
print(word_count(strl))
file=open("words.txt","w",1)
file.write("Number of words in the file are 29")
file.write("""Unique occurences are:
{'a': 1, 'fun': 1, 'horse': 1, 'open': 1, 'it': 1, 'in': 1, 'sleigh': 1, 'is': 1, 'all': 2, 'oh': 1, 'the': 2, 'one': 1, 'to': 1, 'ride': 1, 'jingle': 6, 'way': 2, 'wh
```

## **Assignment 35**

### Assignment 35

Duration

Assume the following Python code:

```
mylist = [1,2,3,"4",5]
sum = 0
for i in mylist:
    sum = sum + i
print(sum)
print(mylist[5])
```

Rewrite the code to handle the exceptions raised. Print appropriate error messages wherever applicable.



assignment35.py - C:\Users\500060720\Desktop\Post Freshmann\Locker\Python Project

File Edit Format Run Options Window Help

```
try:
    mylist=[1,2,3,"4",5]
    sum=0
    for i in mylist:
        sum=sum+i
    print(sum)
except:
    print("Value error found")
    mylist[3]=int(mylist[3])
finally:
    sum=0
    for i in mylist:
        sum=sum+i
    print(sum)
try:
    print(mylist[5])
except:
    print("Index max value exceeded. Overflow encountered")
```

Python 3.5.0 Shell

File Edit Shell Debug Options Window Help

Python 3.5.0 (v3.5.0:374f501f4567, Sep 13 2015, 02:27:37) [AMD64] on win32

Type "copyright", "credits" or "license()" for more information  
>>>

RESTART: C:\Users\500060720\Desktop\Post Freshmann\Locker\assignment35.py

Value error found

15

Index max value exceeded. Overflow encountered

>>> |

## **Assignment 36**

### Assignment 36

Duration : 15 mins

You have already created a Python program to implement the following in file handling section:

- 1.read a file.
- 2.add backslash (\) before every double quote in the file contents.
- 3.write it to another file in the same folder.
- 4.print the contents of both the files.

Modify your code to implement Exception handling. Print appropriate error messages wherever applicable.

```
Python 3.5.0 Shell
File Edit Shell Debug Options Window Help
Python 3.5.0 (v3.5.0:374f501f4567, Sep 13 2015, 02:27:37) [MS
Type "copyright", "credits" or "license()" for more informati
>>>
  RESTART: C:\Users\500060720\Desktop\Post Freshmann\Locker\Py
1) Read a file
Jack said, "Hello Pune"

2) Add backslash\ before every double quote in file contents
Jack said, \"Hello Pune\"

3) Write it to another file in the same folder

4) Write contents of both the files
Jack said, \"Hello Pune\"
Jack said, "Hello Pune"
No errors found
Task successfully done
>>>
```

assignment36.py - C:\Users\500060720\Desktop\Post Freshmann\Locker\Python Project\assignment36.py (3.

File Edit Format Run Options Window Help

```
try:
    file=open("testfile1.txt","r",1)
    print("1)Read a file")
    s=file.read()
    print(s)
    print("")
    print("")
    print("2) Add backslash\ before every double quote in file contents")
    list1=s.split('"')
    str1=str()
    for k in range (0,len(list1)):
        if k!=len(list1)-1:
            str1=str1+list1[k]+'\\'
        else:
            str1=str1+list1[k]
    print(str1)
    print("")
    print("")
    print("3) Write it to another file in the same folder")
    file.close()
    file=open("testfile2.txt","w",1)
    file.write(str1)
    print("")
    print("")
    print("4) Write contents of both the files")
    print(str1)
    print(s)
except TypeError:
    print("Type Error found")
except ValueError:
    print("Value Error found")
except IOError:
    print("Input Output Error found")
except:
    print("Other error")
else:
    print("No errors found")
finally:
    print("Task successfully done")
```

## **Assignment 37**

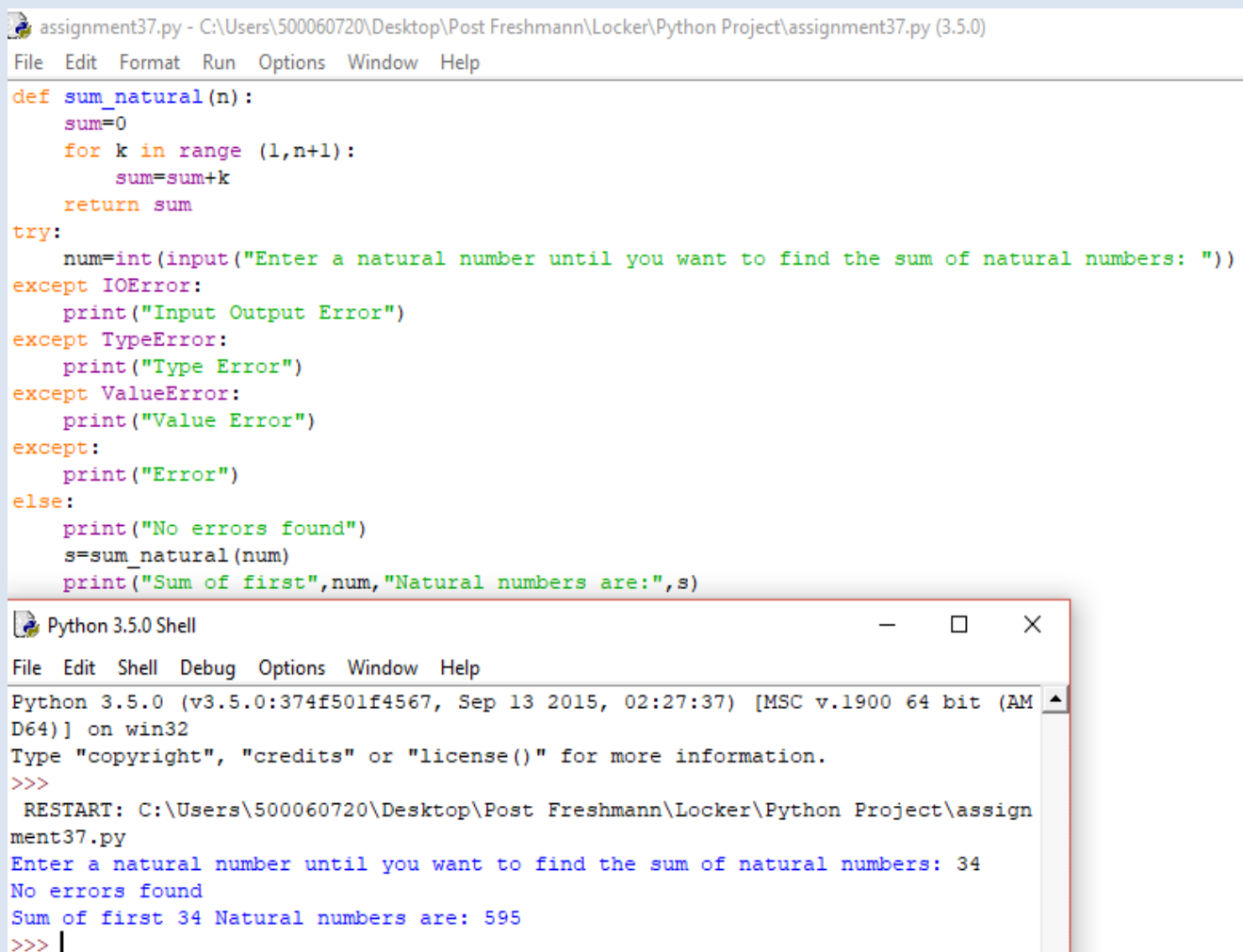
### Assignment 37

Duration : 10 mins

You have already executed the Python program given below in Functions section:

- Add natural numbers up to n where n is taken as an input from user.

Do appropriate exception handling in the code and observe the output by providing invalid input values.



```
assignment37.py - C:\Users\500060720\Desktop\Post Freshmann\Locker\Python Project\assignment37.py (3.5.0)
File Edit Format Run Options Window Help

def sum_natural(n):
    sum=0
    for k in range (1,n+1):
        sum=sum+k
    return sum
try:
    num=int(input("Enter a natural number until you want to find the sum of natural numbers: "))
except IOError:
    print("Input Output Error")
except TypeError:
    print("Type Error")
except ValueError:
    print("Value Error")
except:
    print("Error")
else:
    print("No errors found")
    s=sum_natural(num)
    print("Sum of first",num,"Natural numbers are:",s)

Python 3.5.0 Shell
File Edit Shell Debug Options Window Help
Python 3.5.0 (v3.5.0:374f501f4567, Sep 13 2015, 02:27:37) [MSC v.1900 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:\Users\500060720\Desktop\Post Freshmann\Locker\Python Project\assignment37.py
Enter a natural number until you want to find the sum of natural numbers: 34
No errors found
Sum of first 34 Natural numbers are: 595
>>> |
```

## **Assignment 38**

### Assignment 38

Duration: 15 mins

Refer to the following assignment which you have already executed in Functions section. Modify your code to implement Exception Handling and display appropriate error message wherever applicable.

At an airport, a traveler is allowed entry into the flight only if he clears the following checks:

1. Baggage Check
2. Immigration Check
3. Security Check

The logic for the check methods are given below:

#### **check\_baggage (baggage\_weight)**

- returns True if *baggage\_weight* is greater than or equal to 0 and less than or equal to 40. Otherwise returns False.

#### **check\_immigration (expiry\_year)**

- returns True if *expiry\_year* is greater than or equal to 2001 and less than or equal to 2025. Otherwise returns False.

#### **check\_security(noc\_status)**

- returns True if *noc\_status* is 'valid' or 'VALID', for all other values return False.

#### **traveler()**

- Initialize the traveler Id and traveler name and invoke the functions `check_baggage()`, `check_immigration()` and `check_security()` by passing required arguments.
- Refer the table below for values of arguments.
- If all values of `check_baggage()`, `check_immigration()` and `check_security()` are true,

display *traveler\_id* and *traveler\_name*

display "Allow Traveler to fly!"

Otherwise,

display *traveler\_id* and *traveler\_name*

display "Detain Traveler for Re-checking!"

Variable	Value
<i>traveler_id</i>	1001
<i>traveler_name</i>	Jim
<i>baggage_weight</i>	Thirty
<i>expiry_year</i>	2016
<i>noc_status</i>	0

Invoke the `traveler()` function. Modify the values of different variables in `traveler()` function and observe the output

assignment38.py - C:\Users\500060720\Desktop\Post Freshmann\Locker\Python Project\assignment38.py (3.5.0)

File Edit Format Run Options Window Help

```
def check_baggage(baggage_weight):
    if baggage_weight>=0 and baggage_weight<=40:
        return True
    else:
        return False
def check_immigration(expiry_year):
    if expiry_year>=2001 and expiry_year<=2025:
        return True
    else:
        return False
def check_security(noc_status):
    if noc_status=="valid" or noc_status=="VALID":
        return True
    else:
        return False
def traveler():
    try:
        traveler_id=input("Enter the Traveler ID: ")
        traveler_name=input("Enter the Traveler name: ")
        baggage_weight=int(input("Enter the baggage weight: "))
        expiry_year=int(input("Enter the expiry year of immigration: "))
        noc_status=input("Enter the noc status of the traveler: ")
        print(traveler_id)
        print(traveler_name)
        if check_baggage(baggage_weight)==True and check_immigration(expiry_year)==True and check_security(noc_status)==True:
            print("Allow traveler to fly!")
        else:
            print("Detain traveler for re-checking")
    except:
        print("Error encountered")
traveler()
```

Python 3.5.0 Shell

File Edit Shell Debug Options Window Help

Python 3.5.0 (v3.5.0:374f501f4567, Sep 13 2015)  
Type "copyright", "credits" or "license()" for more

>>>

RESTART: C:\Users\500060720\Desktop\Post Freshmann\Locker\Python Project\assignment38.py

Enter the Traveler ID: 500060720

Enter the Traveler name: Nishkarsh

Enter the baggage weight: 10

Enter the expiry year of immigration: 2020

Enter the noc status of the traveler: valid

500060720

Nishkarsh

Allow traveler to fly!

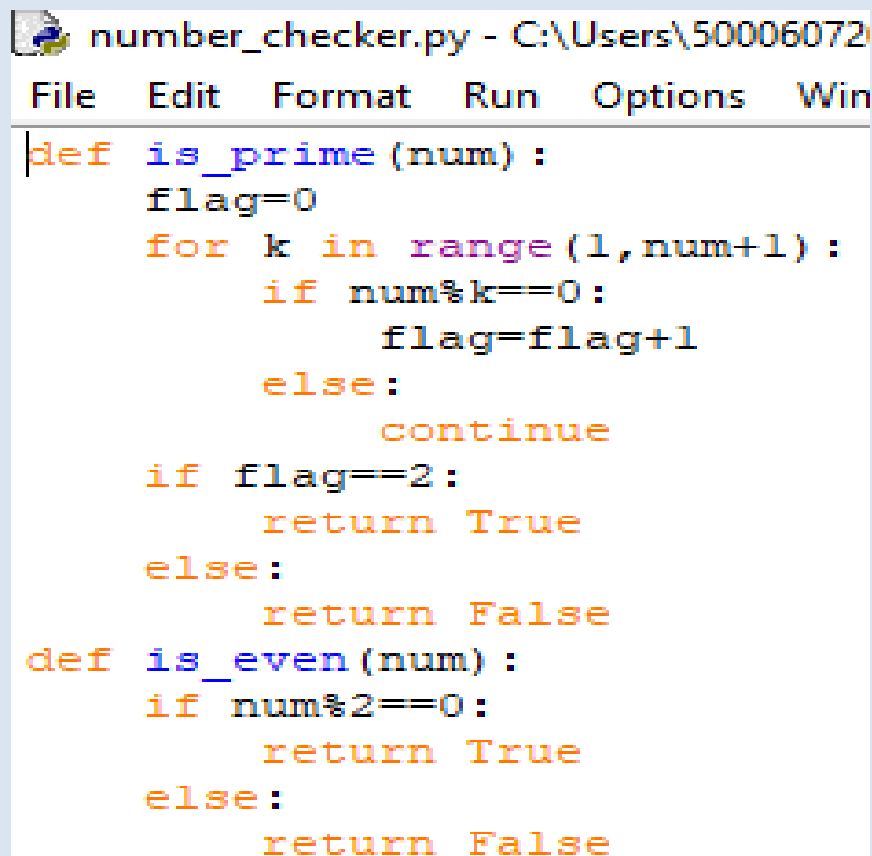
>>> |

## **Assignment 39**

### Assignment 39

- Create a module "number\_checker.py" which has following 2 functions:
  - is\_prime(num) : this function returns true if the input number is prime
  - is\_even(num): this function returns true if the input number is even
- Create another Python module "test\_module.py".
- Invoke the functions "is\_prime(num)" and "is\_even(num)" in "test\_module.py".
- Observe the results.

Hint: Import "number\_checker.py" module in "test\_module.py" before using it's functions.




```
number_checker.py - C:\Users\50006072\
File Edit Format Run Options Win
def is_prime (num) :
    flag=0
    for k in range (1,num+1) :
        if num%k==0:
            flag=flag+1
        else:
            continue
    if flag==2:
        return True
    else:
        return False
def is_even (num) :
    if num%2==0:
        return True
    else:
        return False
```

 test\_module.py - C:\Users\500060720\Desktop\Post Freshmann\Locker\Py

File Edit Format Run Options Window Help

```
import number_checker
num=int(input("Enter a number for checking: "))
print("")
print("Let's check whether number is prime or not")
print("Is prime?:",number_checker.is_prime(num))
print("")
print("")
print("Let's check whether number is even or not")
print("Is even?:",number_checker.is_even(num))
```

 Python 3.5.0 Shell

File Edit Shell Debug Options Window Help

Python 3.5.0 (v3.5.0:374f501f4567, Sep 13 2015, 02:00:00) [AMD64] on win32

Type "copyright", "credits" or "license()" for more

>>>

RESTART: C:\Users\500060720\Desktop\Post Freshmann\Locker\test\_module.py

Enter a number for checking: 283

Let's check whether number is prime or not

Is prime?: True

Let's check whether number is even or not

Is even?: False

>>> |



## **Assignment 40**

### Assignment 40

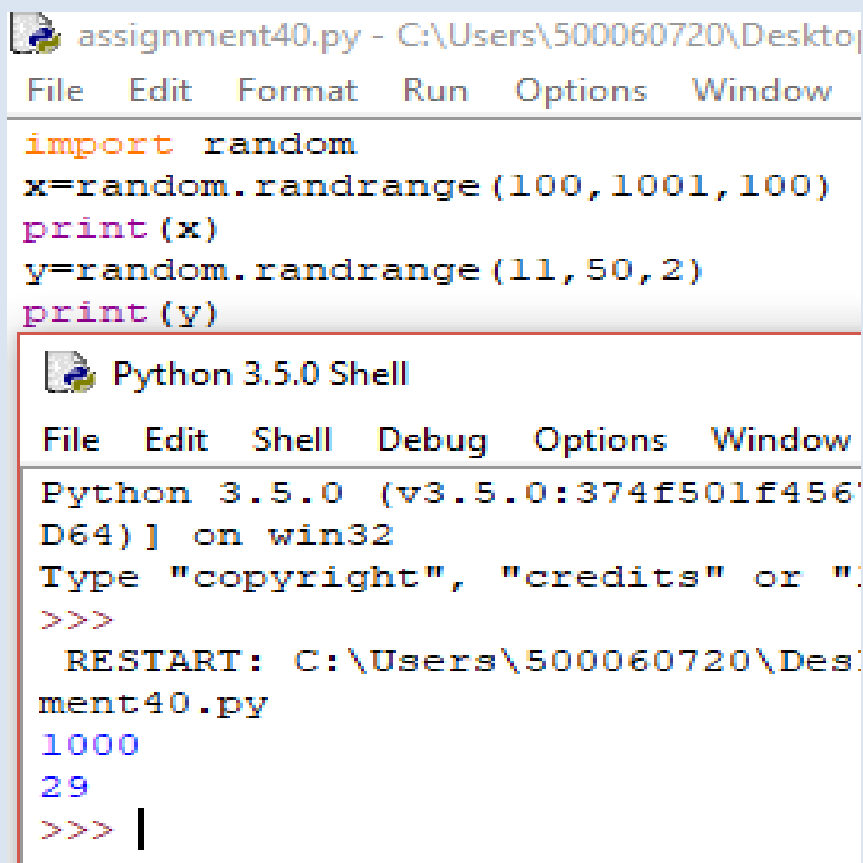
Duration : 10 mins

- Write a Python program to randomly print any of the below numbers:

100,200,300,400,500,600,700,800,900,1000

Execute the program 10 times and verify if the number generated in every output is one out of the numbers given in the list above.

- Write a Python program to print a random odd numbers between 10 and 50.



```
assignment40.py - C:\Users\500060720\Desktop
File Edit Format Run Options Window
import random
x=random.randrange(100,1001,100)
print(x)
y=random.randrange(11,50,2)
print(y)

Python 3.5.0 Shell
File Edit Shell Debug Options Window
Python 3.5.0 (v3.5.0:374f501f4567
D64)] on win32
Type "copyright", "credits" or "
>>>
RESTART: C:\Users\500060720\Des
ment40.py
1000
29
>>> |
```

## **Assignment 41**

### Assignment 41

Duration : 15 mins

Write a Python program for rolling a dice on clicking enter key. The program should run infinitely until user enters 'q'.

Hint:

- To implement a dice, you can randomly print a number in the range 1-6.
- Refer [this](#) video for one of the possible solutions.



```
import random
```

```
leave_program='a'
```

```
print("This is a program for Dice Throw!!!")
```

```
print("Caution! Infinite Loop ahead!!! Type q to quit")
```

```
print("Press Enter to continue")
```

```
input()
```

```
while leave_program != 'q':
```

```
    x=random.randint(1,6)
```

```
    if x==1:
```

```
        print("      ")
```

```
        print("      ")
```

```
        print("    0  ")
```

```
        print("      ")
```

```
        print("      ")
```

```
        print()
```

```
        leave_program=input()
```

```
    if x==2:
```

```
        print("      ")
```

```
        print("      ")
```

```
        print(" 0    0 ")
```

```
        print("      ")
```

```
print("      ")
```

```
print()
```

```
leave_program=input()
```

```
if x==3:
```

```
print("      ")
```

```
print("  0  ")
```

```
print("      ")
```

```
print(" 0  0 ")
```

```
print("      ")
```

```
print()
```

```
leave_program=input()
```

```
if x==4:
```

```
print("      ")
```

```
print(" 0  0 ")
```

```
print("      ")
```

```
print(" 0  0 ")
```

```
print("      ")
```

```
print()
```

```
leave_program=input()
```

```
if x==5:
```

```
print("      ")
```

```
print(" 0  0 ")
```

```
print("    0  ")
```

```
print(" 0  0 ")
```

```
print("      ")
```

```
print()
```

```
leave_program=input()
```

```
if x==6:
```

```
print("      ")
```

```
print(" 0  0 ")
```

```
print(" 0  0 ")
```

```
print(" 0  0 ")
```

```
print("      ")
```

```
print()
```

```
leave_program=input()
```

```
*Python 3.5.0 Shell*
File Edit Shell Debug Options Window Help
Type copyright / license or license / for an
>>>
RESTART: C:\Users\500060720\Desktop\Post Freshm
This is a program for Dice Throw!!!
Caution! Infinite Loop ahead!!! Type q to quit
Press Enter to continue

0 0
0 0
0 0

0 0
0 0
0 0

0 0
0 0

0 0

0
```

## **Assignment 42**

### Assignment 42

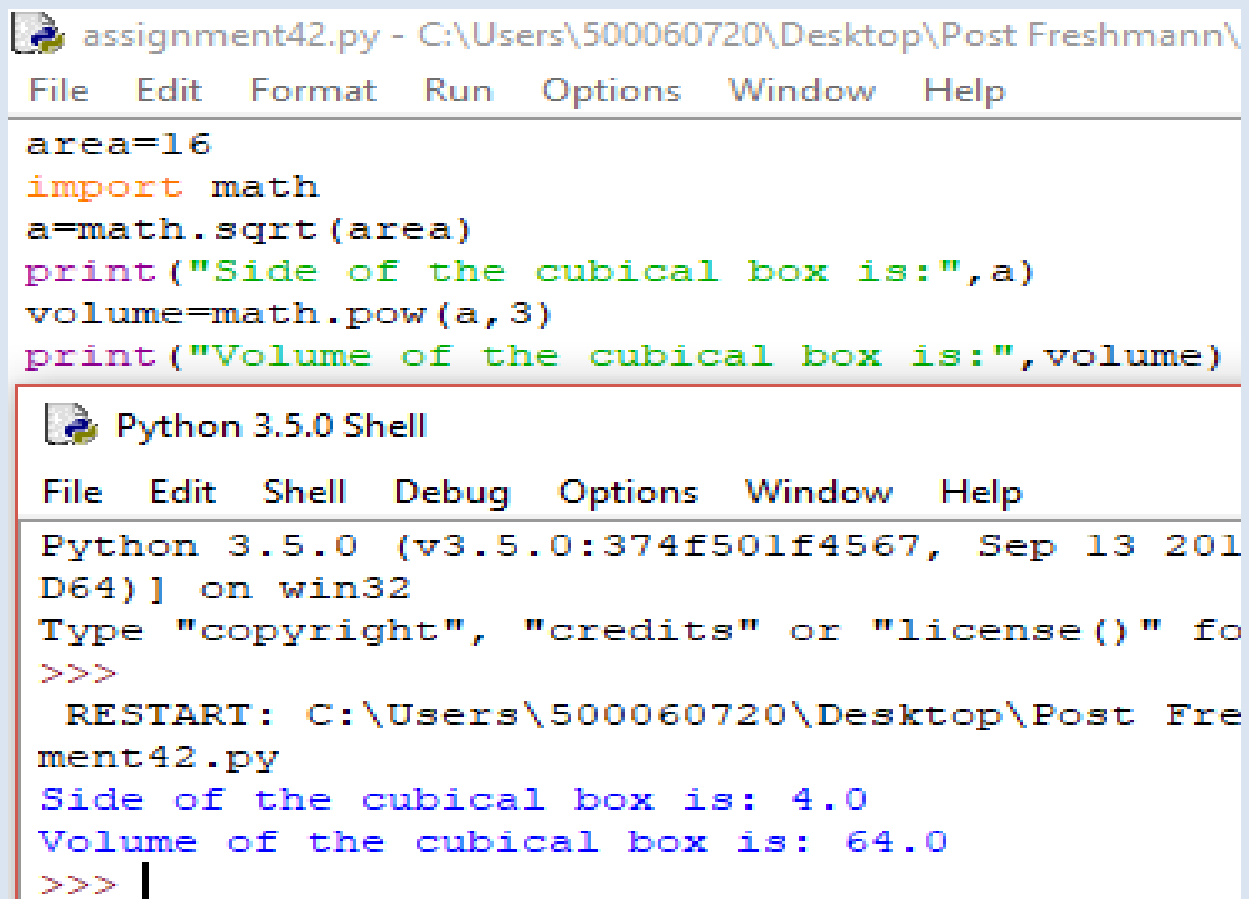
Duration : 10 mins

If area of one wall of a cubical wooden box is 16 units, write a Python program to display the volume of the box.

Note: Area of a cube with side 'a' is 'a\*\*2'.

Volume of the cube can be computed as 'a\*\*3'.

Hint: Make use of 'sqrt' and 'pow' functions from math module.



```
assignment42.py - C:\Users\500060720\Desktop\Post Freshmann\
File Edit Format Run Options Window Help
area=16
import math
a=math.sqrt(area)
print("Side of the cubical box is:",a)
volume=math.pow(a,3)
print("Volume of the cubical box is:",volume)

Python 3.5.0 Shell
File Edit Shell Debug Options Window Help
Python 3.5.0 (v3.5.0:374f501f4567, Sep 13 201
D64)] on win32
Type "copyright", "credits" or "license()" fo
>>>
RESTART: C:\Users\500060720\Desktop\Post Fre
ment42.py
Side of the cubical box is: 4.0
Volume of the cubical box is: 64.0
>>> |
```

## **Assignment 43**

### Assignment 43

Duration : 10 mins

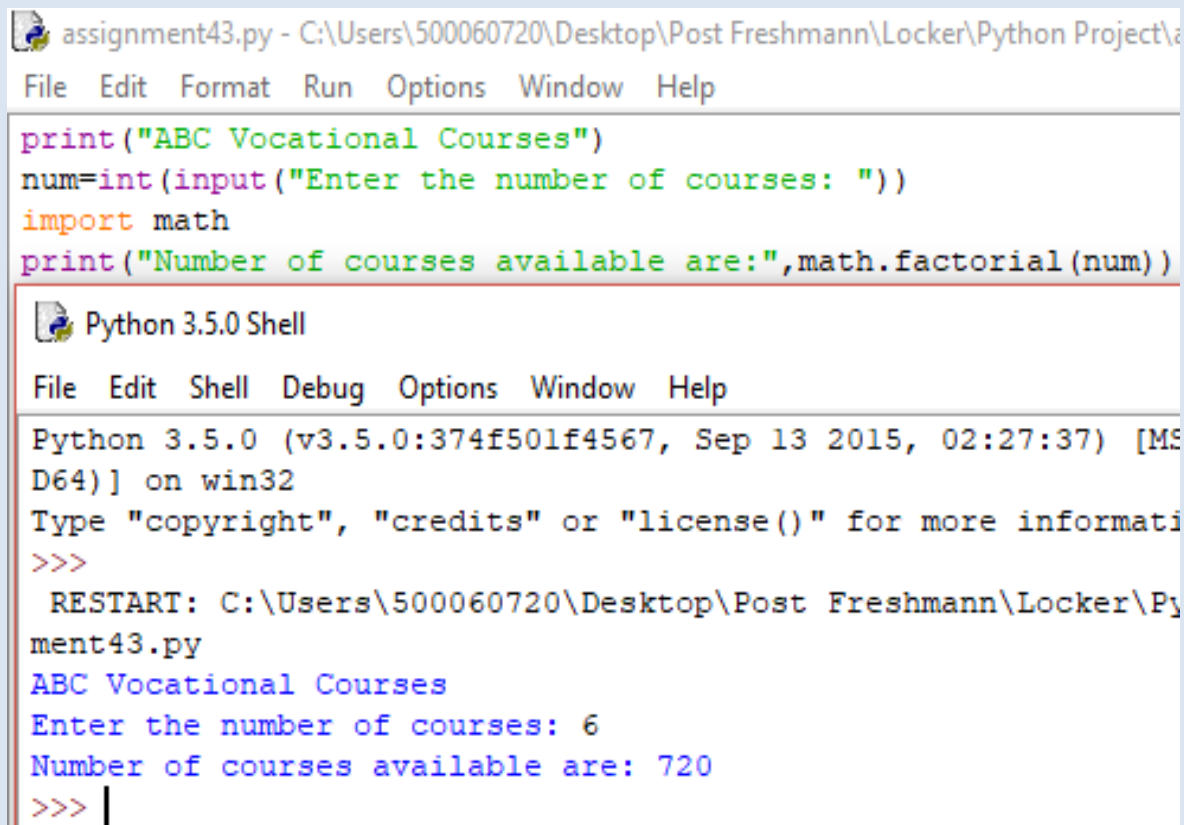
The ABC Institute offers vocational courses to students in multiple areas e.g. theatre, classical singing, traditional dance forms, Bollywood dance, literature and so on. A student can enroll for zero to all courses.

Write a Python function that takes the number of courses as an input and returns the total number of different course combinations, a student can opt for. (Make use of functions available in math module)

Hint:

if no\_of\_courses = 2, possible number of combinations are 2! i.e. 2

if no\_of\_courses = 3, possible number of combinations are 3! i.e. 6 and so on.



```
assignment43.py - C:\Users\500060720\Desktop\Post Freshmann\Locker\Python Project\
File Edit Format Run Options Window Help

print("ABC Vocational Courses")
num=int(input("Enter the number of courses: "))
import math
print("Number of courses available are:",math.factorial(num))

Python 3.5.0 Shell
File Edit Shell Debug Options Window Help
Python 3.5.0 (v3.5.0:374f501f4567, Sep 13 2015, 02:27:37) [MS
D64)] on win32
Type "copyright", "credits" or "license()" for more informati
>>>
RESTART: C:\Users\500060720\Desktop\Post Freshmann\Locker\Py
ment43.py
ABC Vocational Courses
Enter the number of courses: 6
Number of courses available are: 720
>>> |
```



## **Assignment 44**

### Assignment 44

Execute the following code and observe the output.

```
1. import time
2. print(time.time())
3. print(time.localtime())
4. print(time.localtime(time.time()))
5. print(time.asctime())
6. mytime = (2016,7,27,15,45,23,0,0,0)
7. print(time.localtime(time.mktime(mytime)))
```

assignment44.py - C:\Users\500060720\Desktop\Post Freshmann\Locker\Python Project\assignment44.py (3.5.0)

File Edit Format Run Options Window Help

```
import time
print(time.time())
print(time.localtime())
print(time.localtime(time.time()))
print(time.asctime())
mytime=(2016,7,27,15,45,23,0,0,0)
print(time.localtime(time.mktime(mytime)))
```

Python 3.5.0 Shell

File Edit Shell Debug Options Window Help

Python 3.5.0 (v3.5.0:374f501f4567, Sep 13 2015, 02:27:37) [MSC v.1900 64 bit (AMD64)] on win32

Type "copyright", "credits" or "license()" for more information.

>>>

RESTART: C:\Users\500060720\Desktop\Post Freshmann\Locker\Python Project\assignment44.py

1529070311.2262888

time.struct\_time(tm\_year=2018, tm\_mon=6, tm\_mday=15, tm\_hour=19, tm\_min=15, tm\_sec=11, tm\_wday=4, tm\_yday=166, tm\_isdst=0)

time.struct\_time(tm\_year=2018, tm\_mon=6, tm\_mday=15, tm\_hour=19, tm\_min=15, tm\_sec=11, tm\_wday=4, tm\_yday=166, tm\_isdst=0)

Fri Jun 15 19:15:11 2018

time.struct\_time(tm\_year=2016, tm\_mon=7, tm\_mday=27, tm\_hour=15, tm\_min=45, tm\_sec=23, tm\_wday=2, tm\_yday=209, tm\_isdst=0)

>>> |

## **Assignment 45**

### Assignment 45

Duration : 15 mins

Consider a Python string:

```
cust_details = "Hello John, your customer id is j181"
```

- 1) Find, if the name of the customer is preceded by a pattern "Hello " or "hello " (Observe a space after the word)? If pattern is found, print the searched result.
- 2) Find, if the given string ends with a pattern containing only one alphabet followed by three numbers? If pattern is found, print the searched result.
- 3) Replace the word starting with "j" followed by three numbers to only the number(remove the alphabet).
- 4) Replace the word "id" with "ID".

The output of the above code is "Hello John, your customer ID is 181"

Hint:

In questions 1 and 2 use search() function. The matched object can be returned using group() function of match object.

Use sub() function in question 3 and 4.

```
Python 3.5.0 Shell
File Edit Shell Debug Options Window Help
Python 3.5.0 (v3.5.0:374f501f4567, Sep 13 2015, 02:27:37) [MSC v.1900 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:\Users\500060720\Desktop\Post Freshmann\Locker\Python Project\assignment45.py
1 Find if the customer name is preceded by hello or Hello and mention if yes
The name is preceded by Hello or hello as: Hello

2) Find, if the given string ends with a pattern containing only one alphabet followed by three numbers? If pattern is found, print the searched result
The query is true and has result: j181

3) Replace the word starting with "j" followed by three numbers to only the number(remove the alphabet).
Hello John, your customer id is 181

4) Replace the word starting with "j" followed by three numbers to only the number(remove the alphabet).
Hello John, your customer ID is 181
>>> |
```

File Edit Format Run Options Window Help

```

cust_details="Hello John, your customer id is j181"
import re
print("1 Find if the customer name is preceded by hello or Hello and mention if yes")
match=re.search("hello |Hello ",cust_details,re.I)
if (match.group()!=None):
    print("The name is preceded by Hello or hello as:",match.group())
else:
    print("Name is not greeted as asked")
print("")
print("")
print("2) Find, if the given string ends with a pattern containing only one alphabet followed by three numbers? If pattern is found, print the searched result")
match=re.search("\D\d{3}",cust_details)
if match.group()!=None:
    print("The query is true and has result:",match.group())
else:
    print("Query is not true")
print("")
print("")
print("3) Replace the word starting with "j" followed by three numbers to only the number(remove the alphabet).")
cust_details=re.sub("j181","181",cust_details)
print(cust_details)
print("")
print("")
print("4) Replace the word starting with "j" followed by three numbers to only the number(remove the alphabet).")
cust_details=re.sub("id","ID",cust_details)
print(cust_details)

```

## **Assignment 46**

### Assignment 46

Duration : 20 mins


Consider a scenario of managing student details in ABC Training Institute. Write a Python program to implement the business requirements mentioned below:

- a) Accept student\_id and validate whether it contains only digits.
- b) If student\_id is valid, accept student\_name from the user and validate whether it contains only alphabets.
- c) If student\_name is valid, accept fees\_amount paid by the student:
  - 1. Decimal point is optional in fees\_amount(can have maximum one decimal point)
  - 2. Only two digits are allowed after decimal point
- d) If invalid data is entered in any of the above steps, display appropriate error messages. Else, create an email\_id for student as [student\\_name@ABC.com](mailto:student_name@ABC.com). Assume there are no duplicate names.
- e) Perform above validations using Regular Expressions and print details of the student: student\_id, student\_name, fees\_amount, email\_id

assignment46.py - C:\Users\500060720\Desktop\Post Freshmann\Locker\Python Project\assignment46.py (3.5.0)

File Edit Format Run Options Window Help

```
institute="ABC Training Institute"
student_id=(input("Enter the student ID: "))
import re
email_id=str() #empty string declaration
if re.search("^\d+$",student_id):
    student_name=input("Enter name of the Student: ")
    if re.search("^[a-zA-Z]+$",student_name): #Question says only alphabets and thus space is not allowed in check so name can be only one word
        fees_amount=input("Enter the fees to be paid as a number with atmost two precision after decimal: ")
        if re.search("^\d+\.\d{2}$",fees_amount):
            if fees_amount.count(".")>1:
                print("An amount cannot have more than one period symbol")
            else:
                email_id=student_name+"@ABC.com"
        else:
            print("Wrong Format for fees")
    else:
        print("Name must have only alphabets in it!!!")
else:
    print("Enter student id only as combination of digits!!!")
print("Details of the Student are as follows:")
print("Student ID:",student_id)
print("Student Name:",student_name)
print("Fees Paid:",fees_amount)
print("Email ID:",email_id)
```

 Python 3.5.0 Shell

File Edit Shell Debug Options Window Help

Python 3.5.0 (v3.5.0:374f501f4567, Sep 13 2015, 02:27:37) [MSC v.1900 64 bit (AMD64)] on win32

Type "copyright", "credits" or "license()" for more information.

>>>

RESTART: C:\Users\500060720\Desktop\Post Freshmann\Locker\Python Project\assignment46.py

Enter the student ID: 500060720

Enter name of the Student: Nishkarsh

Enter the fees to be paid as a number with atmost two precision after decimal: 500.34

Details of the Student are as follows:

Student ID: 500060720

Student Name: Nishkarsh

Fees Paid: 500.34

Email ID: Nishkarsh@ABC.com

>>> |

## **Assignment 47**

### Assignment 47

Duration : 10 mins

Consider a string:

```
my_string = """Strings are amongst the most popular data types in Python. We can create the strings by enclosing characters in quotes. Python treats single quotes the same as double quotes."""
```

1) Write a Python program to count the number of occurrences of word "String" in the given string ignoring the case.

2) Write a function "count\_words" to print the count of occurrences of a word:

a) which end with "on". (e.g. Python)

b) which have "on" in between the first and last characters (e.g. amongst)



assignment47.py - C:\Users\500060720\Desktop\Post Freshmann\Locker\Python Project\assignment47.py (3.5.0)

File Edit Format Run Options Window Help

```
my_string="""Strings are amongst the most popular data types in Python. We can create the strings by enclosing cha
print("The number of occurrences of word string are:",my_string.lower().count("string"))
def count_words(strl):
    listl=strl.split()
    for k in range(0,len(listl)):
        if listl[k].endswith("on")==True:
            print("Occurrences of word",listl[k],"with on at end are:",strl.count(listl[k]))
        else:
            continue
    for k in range(0,len(listl)):
        if (listl[k].startswith("on")!=True) and (listl[k].endswith("on")!=True) and (listl[k].find("on")!=(-1)):
            print("Occurrences of word",listl[k],"with on in between are:",strl.count(listl[k]))
        else:
            continue
count_words(my_string)
```

Python 3.5.0 Shell

File Edit Shell Debug Options Window Help

Python 3.5.0 (v3.5.0:374f501f4567, Sep 13 2015, 02:27:37) [MSC v.1900 64 bit (AMD64)] on win32

Type "copyright", "credits" or "license()" for more information.

>>>

RESTART: C:\Users\500060720\Desktop\Post Freshmann\Locker\Python Project\assignment47.py

The number of occurrences of word string are: 2

Occurrences of word Python with on at end are: 2

Occurrences of word amongst with on in between are: 1

Occurrences of word Python. with on in between are: 1

>>>

## **Assignment 48**

### Assignment 48

Consider the price list of various items in the Retail Store:

*item\_price = [1050, 2200, 8575, 485, 234, 150, 399]*

Customer John wants to know the:

1. Price of costliest item sold in retail store
2. Number of items in the Retail store
3. Prices of items in increasing order
4. Prices of items in descending order

Implement the above mentioned business requirements using built-in List functions.

Hint -

- 1) Use max, len, sort functions of math module.
- 2) For question 3 and 4, display the list in increasing/decreasing order.

assignment48.py - C:\Users\500060720\Desktop\Post Freshmann\Locker\Python Project\assignment48.py (3.5.0)

File Edit Format Run Options Window Help

```
print("Retail Store Prices")
item_price=[1050,2200,8575,485,234,150,399]
customer_name="John"
print("Hello",customer_name)
print("1) Price of the costliest item sold in Retail Store is:",max(item_price))
print("2) Number of items in Retail Store are:",len(item_price))
#print("3) Prices of items in increasing order are:",item_price.sort())
#print("4) Prices of items in decreasing order are:",item_price.sort().reverse())
print("3) Prices of items in increasing order are:",(sorted(item_price)))
print("4) Prices of items in decreasing order are:",(sorted(item_price, reverse=True)))
```

Python 3.5.0 Shell

File Edit Shell Debug Options Window Help

Python 3.5.0 (v3.5.0:374f501f4567, Sep 13 2015, 02:27:37) [MSC v.1900 64 bit (AMD64)]  
32

Type "copyright", "credits" or "license()" for more information.

>>>

RESTART: C:\Users\500060720\Desktop\Post Freshmann\Locker\Python Project\assignment48

Retail Store Prices

Hello John

1) Price of the costliest item sold in Retail Store is: 8575

2) Number of items in Retail Store are: 7

3) Prices of items in increasing order are: [150, 234, 399, 485, 1050, 2200, 8575]

4) Prices of items in decreasing order are: [8575, 2200, 1050, 485, 399, 234, 150]

>>>