```
        self.x=x
    def area (self):
        print('Area of square is=',self.x*self.x)
class rectangle(square):
    def __init__(self,x,y):
        super().__init__(x)
        self.y=y
    def area(self):
        super().area()
        print('Area of Rectangle is=',self.x*self.y)
a,b=[float(x) for x in input('Enter TWO measures:').split()]
r=rectangle(a,b)
r.area()
```

```
Enter TWO measures:5 7
Area of square is= 25.0
Area of Rectangle is= 35.0
```

```
# A python program to overload greater that(>) operator to make it act on class objects
#gt-(>), lt-(<)

class Ramayan:
    def __init__(self,pages):
        self.pg = pages
    def __gt__(self,other):
        return self.pg>other.pg

class Mahabharat:
    def __init__(self,pages):
        self.pg=pages
Ram = Ramayan(1200)
Mah = Mahabharat(2000)

if (Ram>Mah):
    print('Ramayan has more pages')
else:
    print('Mahabharat has more pages')
```

```
Mahabharat has more pages
```

DAY -30 SEP- 9

```
# REULAR EXPRESSION

import re

# SEARCH
```

```
str='Apple Mango Pappya'
s= re.search(r'Mango',str)
print(s)
if s:
    print(s.group())
```

```
<re.Match object; span=(6, 11), match='Mango'>
Mango
```

```
# START OF STRiNG(^) - REGULAR EXPRESSION
str='Straive company looks good'
y=re.search(r'^Straive',str)

if y:
    print(y,"\n Starts with 'Straive'")
else:
    print(y,"\n Does not starts with 'Straive'")
```

```
<re.Match object; span=(0, 7), match='Straive'> Starts with 'Straive'
```

```
str='Straive company looks good'
y=re.search(r'^StraivE',str)

if y:
    print(y,"\n Starts with 'Straive'")
else:
    print(y,"\n Does not starts with 'Straive'")
```

```
None Does not starts with 'Straive'
```

```
# END OF STRING($) - REGULAR RXPRESSION

str='Straive company looks good'
y=re.search(r'good$',str)

if y:
    print(y,"\n Ends with 'good'")
else:
    print(y,"\n Does not Ends with 'good'")
```

```
<re.Match object; span=(22, 26), match='good'> Ends with 'good'
```

```
# SEARCH USING IGNORECASE METHOD

str='Malabar Gold & Diamonds'
igr=re.search(r'^malabar',str,re.IGNORECASE)
if igr:
    print(igr,"\n str starts with 'Malabar'")
```

```
else:
    print(igr,"\n str does not starts with 'Malabar'")
```

```
<re.Match object; span=(0, 7), match='Malabar'>
 str starts with 'Malabar'
```

```
# A python program to create a regular expression to retrive marks and
names from a given string
# {2} - number occurence
#\d- decimal, \n - newe line, \r - read
# findall- brings particular charecters

str='Jk got 85 points, Jin got 84 points and the winner of the game is
V he got 98 points'
points=re.findall('\d{2}',str)
print(points)


names=re.findall('[A-Z][a-z]*',str)
print(names)
```

```
['85', '84', '98']
['Jk', 'Jin', 'V']
```

```
str='The programm will be held on Dec 7th at 5pm or 6pm or 8PM or 8AM'
t=re.findall(r'\dpm | \dAM',str)
print(t)
```

```
['5pm ', '6pm ', ' 8AM']
```

DAY - 31 SEP - 10

```
# REGULAR EXPRESSION

# r - RAW INPUT
import re
s= 'English Tamil'
r=re.findall(r'^En',s)
print(r)
```

```
['En']
```

```
s= 'English Tamil'
r=re.findall(r'^EN',s)
print(r)
```

```
[]
```

```python
s= 'English Tamil'
r=re.findall(r'^En',s)
if r:
    print("The string starts with 'En'")
else:
    print("The string does not starts with 'En'")
```

```
The string starts with 'En'
```

```python
s= 'English Tamil'
r=re.findall(r'l$',s)
if r:
    print("The ends starts with 'l'")
else:
    print("The string does not ends with 'l'")
```

```
The ends starts with 'l'
```

```python
# A python program to create a regular expression to search whether a given string is ending with particular word or not

s='Hair oil'
r=re.search(r'oil',s)
if r:
    print(r,"\n The particular string is Available")
else:
    print(" The particular string id Not Available")
```

```
<re.Match object; span=(5, 8), match='oil'>
 The particular string is Available
```

```python
# \b - REPRESENTS THE BEGINING AND ENDING OF THE STRING
pattern=re.compile(r'\b\w{5}\b')
r=pattern.findall('Apple,Mango,Watermelon,Grape')
print(r)
```

```
['Apple', 'Mango', 'Grape']
```

```python
pattern=re.compile(r'\b\w{5}\b')
r=pattern.findall('Apple,Mango,Water melon,Grape')
print(r)
```

```
['Apple', 'Mango', 'Water', 'melon', 'Grape']
```

```python
pattern=re.compile(r'\b\w{10}\b')
r=pattern.findall('Apple,Mango,Watermelon,Grape')
print(r)
```

```
['Watermelon']
```

```python
# A python program to create a regular expression to search at the
ending a string by ignoring the case

from re import*

s='Dates Almonds'
r=search(r'almonDs$',s,IGNORECASE)
if r:
    print(r,"\n string ends with 'Almonds'")
else:
    print(r,"\n string does ends with 'Almonds'")
```

```
<re.Match object; span=(6, 13), match='Almonds'>
 string ends with 'Almonds'
```

```python
# Python program to creat a regex that reads - Email id's from a text
file
import re

fs=open('D:\\python programming\\Regex.txt','r')
for line in fs:
    rs=re.search(r'\s+@\s.+',line)
    print(rs)
fs.close()
```

```
None
None
None
```

```python
# Python program to create a regular expression to replace a string
with a new string

import re
s='My mother tongue is Malayalam'
r=re.sub(r'Malayalam','Telugu',s)
print(r)
```

```
My mother tongue is Telugu
```

```python
print(re.search("Biscut","Oreo Biscuts taste better than Good day
Biscut"))
print(re.match("Biscut","Biscuts taste's good"))
print(re.findall("Biscut","Oreo Biscuts taste better than Good day
Biscut"))
for match in re.finditer("Biscut","Oreo Biscuts taste better than Good
day Biscut"):
    print(match)
```

```
<re.Match object; span=(5, 11), match='Biscut'>
<re.Match object; span=(0, 6), match='Biscut'>
['Biscut', 'Biscut']
<re.Match object; span=(5, 11), match='Biscut'>
<re.Match object; span=(40, 46), match='Biscut'>
```

```
# DISPLAY RAW STRING
# \t - tab space , \n - new line

print("\t Look at there.there is a \n SQUIRREL")
print(r"\t Look at there.there is a \n SQUIRREL")
```

```
	 Look at there.there is a 
 SQUIRREL
\t Look at there.there is a \n SQUIRREL
```