1) Exceptions are errors and are raised when the program is syntactically correct but the code resuts into an error.
Logical Errors can be handled through exceptions.

CODE :-

```
num1 = int(input("Enter 1st number : "))
num 2 = int (input (" Enter 2nd number : "))
try:
        res = num1/num2
except ArithmeticError:
            print (" Value cannot be divided by 0")
else :
        print (res)
finally :
        print (" Arithmetic error is excepted")
```

OUTPUT:

```
  Enter 1st number : 10
  Enter 2nd number : 0
```

Value cannot be divided by 0

Arithmetic Error is excepted.

2)
```
class Rectangle:
    def __init__(self, length, width):
        self.length = length
        self.width = width

    def perimeter(self):
        return 2*(self.length + self.width)

    def area(self):
        return self.length * self.width


    def display(self):
        print("Length of rectangle:", self.length)
        print("Width of rectangle:", self.width)
        print("Perimeter of rectangle:", self.perimeter())
        print("Area of rectangle:", self.area())
class Parallelepipede(Rectangle):
    def __init__(self, length, width, height):
        Rectangle.__init__(self, length, width)
        self.height = height
        def volume(self):
            return self.length * self.width * self.height

obj1 = Rectangle (5,10)
obj1. display ()

obj2 = Parallelepipede (5,10,2)
print("Volume : ", obj2. volume ())
```

OUTPUT :

```
Length of rectangle : 5
Width of rectangle : 10
Perimeter of rectangle : 30
Area of rectangle : 50
Volume : 100
```

3) a)

```
import re
msg = "https://www.washingtonpost.com/news/
        footballinsider/wp/2016/09/02/odell-
        beckhams-fame-rests-on-one-
        stupid-little-ball-josh-norman-tells-
        author/"
```

```
p= r'\d{4}/\d{2}/\d{2}'
re.findall(p, msg)
```

OUTPUT:

```
['2016/09/02']
```

b)
```
import re
msg = "The following example creates an
       ArrayList with a capacity of 50 elements.
       Four elements are then added to the
       ArrayList and the ArrayList is trimmed
       accordingly."
p = r'\b[a|e][\w]+\b'
re.findall(p, msg)
```

OUTPUT:

```
['example', 'an', 'elements', 'elements', 'are',
 'added', 'and', 'accordingly']
```

c)
```
import re
msg = "This is a Road"
old_str = "Road"
new_str = "Rd"
```

```
re.sub(old_str, new_str, msg)
```

OUTPUT:

```
'This is a Rd'
```