1) Create 2 classes for single inheritance named respectively A(base class) and B(derived class)

2) Create 3 data members in class A: a(private), b(protected) and c(public) initialise their values in a parameterized constructor

3) Create a method known as display in both the classes, to display the values of a,b and c

4) While accessing the private member an exception should be raised and a personalized message should be displayed and the exception should be handled properly so that the rest of the code can get executed.

Ans:- Here's an example implementation of the classes A and B with the requested features:-

Inheritance with Exception Handling:-

```
class A:
    def __init__(self, a, b, c):
        self.a = a
        self.b = b
        self.c = c

    def display(self):
        print("Values in Class A:")
        print("a:", self.a)
        print("b:", self.b)
        print("c:", self.c)


class B(A):
    def __init__(self, a, b, c):
        super().__init__(a, b, c)

    def display(self):
        try:
            print("Values in Class B:")
            print("a:", self.a)
        except Exception as e:
```

```
            print("Exception:", e)
            print("Cannot access private member 'a'")
        print("b:", self.b)
        print("c:", self.c)

obj_b = B(1, 2, 3)
obj_b.display()
```

Output:-

Values in Class B:

a: 1

b: 2

c: 3

Explaintion of this code:-

1. In this code, class A represents the base class with private member **a**, protected member **b**, and public member **c**. The values of these members are initialized using a parameterized constructor **__init__()**.

2. Both classes, A and B, have a method named **display()** to display the values of the members. In class B, when accessing the private member **a**, an exception is raised using a try-except block. The personalized exception message is displayed, and the rest of the code continues execution.

3. The code above demonstrates the usage of these classes by creating an object of class B (**obj_b**) and calling its **display()** method.