

## 8.Access Modifiers

1. Create a class with PRIVATE fields, private method and a main method. Print the fields in main method. Call the private method in main method.

Create a sub class and try to access the private fields and methods from sub class.

2. Create a class with PROTECTED fields and methods. Access these fields and methods from any other class in the same package.

Also, Access the PROTECTED fields and methods from child class located in a different package

Access the PROTECTED fields and methods from any class in different package

3. Create a class with PUBLIC fields and methods.

Access the public methods and fields from any class in the same package or different package.



main.py

Shell



```
1  # 1st program...
2  class MyClass:
3
4      __name = "Santhosh Kumar"
5      __age = 25
6
7      def __private_method(self):
8          print("This is a private
9              method")
10
11     def main(self):
12         print("Name:", self.__name)
13         print("Age:", self.__age)
14         self.__private_method()
15
16 if __name__ == "__main__":
17     my_class = MyClass()
18     my_class.main()
19
```



main.py

Shell



Name: Santhosh Kumar

Age: 25

This is a private method

> |



main.py

Shell



```
1  # 2nd program...
2  class BaseClass:
3      def __init__(self, name):
4          self._name = name
5      def get_name(self):
6          return self._name
7      def set_name(self, name):
8          self._name = name
9      def protected_method(self):
10         print("This is a protected
            method")
11  class ChildClass(BaseClass):
12
13      def __init__(self, name):
14          super().__init__(name)
15
16      def access_protected_method(self):
17          self.protected_method()
18  base_class = BaseClass("Santhosh Kumar"
    )
19  print(base_class._name)
20  base_class._name = "Santhosh Kumar"
21  base_class.protected_method()
22  child_class = ChildClass("Peter Parker"
    )
23  child_class.access_protected_method()
24
```



main.py

Shell



Santhosh Kumar

This is a protected method

This is a protected method

> |



main.py

Shell



```
1  # 3rd program...
2  class MyClass:
3      public_field = "This is a public
        field."
4
5      def public_method(self):
6          return "This is a public method
            ."
7  def main():
8      my_class = MyClass()
9      print(my_class.public_field)
10     print(my_class.public_method())
11  if __name__ == "__main__":
12     main()
13
```



main.py

Shell



This is a public field.

This is a public method.

> |