



Getters and Setters – Important Notes



1. What Are Getters and Setters?

- **Getter** = A method to **retrieve** (get) the value of a private variable.
 - **Setter** = A method to **set/change** the value of a private variable.
 - These are used in **object-oriented programming (OOP)** to **protect and control access** to class data.
-



2. Why Use Them?

Reason	Purpose
 Encapsulation	Hide internal data and expose it safely
 Validation	Add rules before setting a value
 Flexibility	Change implementation without changing the interface
 Testing & Logging	Add logging/debugging/monitoring while accessing variables
 Abstraction	Expose only what's necessary, keeping other parts hidden

3. Basic Syntax (Python)

```
class Person:

    def __init__(self):
        self.__name = "Kiran"

    def get_name(self):    # Getter
        return self.__name

    def set_name(self, name): # Setter
        self.__name = name

p = Person()
p.set_name("Mayank")
print(p.get_name())
```

📌 4. Python @property Decorators (Clean Way)

python

CopyEdit

```
class Student:
```

```
    def __init__(self):
```

```
        self.__age = 0
```

```
    @property
```

```
    def age(self):
```

```
        return self.__age
```

```
    @age.setter
```

```
    def age(self, value):
```

```
        if value >= 0:
```

```
            self.__age = value
```

```
        else:
```

```
            raise ValueError("Invalid age")
```

```
s = Student()
```

```
s.age = 18    # Calls setter
```

```
print(s.age)  # Calls getter
```