

## 4. Vehicle Rental System

Program :

```
from abc import ABC, abstractmethod
```

# Abstraction

```
class Vehicle(ABC):
```

```
    def __init__(self, days):
```

```
        self.days = days
```

```
    @abstractmethod
```

```
    def calculate_rent(self):
```

```
        pass
```

# Inheritance

```
class Car(Vehicle):
```

```
    def __init__(self, days):
```

```
        super().__init__(days)
```

```
        self.rate = 1000 # rent per day
```

# Polymorphism

```
    def calculate_rent(self):
```

```
        return self.days * self.rate
```

```
class Bike(Vehicle):
```

```
def __init__(self, days):
    super().__init__(days)
    self.rate = 500 # rent per day

# Polymorphism
def calculate_rent(self):
    return self.days * self.rate

# ---- Main Program ----

vehicle_type = input("Enter Vehicle (Car/Bike): ")
days = int(input("Enter Number of Days: "))

if vehicle_type.lower() == "car":
    v = Car(days)
elif vehicle_type.lower() == "bike":
    v = Bike(days)
else:
    print("Invalid Vehicle Type")
    exit()

print("Total Rent:", v.calculate_rent())
```

Output :

Enter Vehicle (Car/Bike): Car

Enter Number of Days: 30

Total Rent: 30000