

```
#####
#####
#####
# Defining classes
#####
#####
#####

class CARPARK:
    def __init__(self,buildingname,parkingslot,price):
        self.bname=buildingname
        self.slot=parkingslot
        self.price=price

#####
#####
#####
# Database
#####
#####
#####

carpark1=CARPARK("NCP London Bloomsbury Square Car Park",168,"$10")
carpark2=CARPARK("Waterloo Car Park",153,"$10")
carpark3=CARPARK("The Mayfair Car Park",124,"$15")

#####
#####
#####
# Parking System Logic
#####
#####
#####

def detectobject():
    objectlist=["empty","dynamic","static"]
    obj=999
    while obj > len(objectlist):
        obj=int(input("Scenario: Any obstdacle in the slot now [0:Empty 1:Moving
object (Human/Animal) 2:Static object]?"))
    return objectlist[obj]

#####
#####
#####
# Main program
# Step 1: Choose carpark
```

```

#          Variable used: icp, j, pk, cp
#####
#####
#####

for icp in range(30): print("\n")
carparklist=[carpark1, carpark2, carpark3]
j=0
for pk in carparklist:
    print(j, pk.bname, "-", pk.price,"per hour")
    j=j+1

cp=999
while cp > j:
    print("=====")
    cp=int(input("Please select the carpark ?"))
    print("=====")
selectpark=carparklist[cp]
print("=====")
print("\nBooking selected carpark at ",selectpark.bname)
print("The car is going to ",selectpark.bname)
print("The car arrived the carpark slot", selectpark.slot)
print("=====")

obstdacle_object = False
while obstdacle_object != "empty" :
    obstdacle_object = detectobject()
    if obstdacle_object == "dynamic" :
        print("=====")
        print("The carpark slot has a person, an animal, or anything else, wait for clearing")
        print("=====")
    if obstdacle_object == "static":
        print("=====")
        print("Call building management and notice the driver. Wait for clearing")
        print("=====")

print("=====")
print("Car is parked. Informed the driver")
print("Program end")
print("=====")
print("=====")

```