Sakshi Divakar-

Ass of day 5-

```
class Account:
def _init_(self, owner, balance=0):
self.owner = owner
self.balance = balance
def _str_(self):
return "Account owner: Pavan \nAccount balance: 100"
def deposit(self, dep_amt):
self.balance += dep_amt
print("Deposit Accepted")
def widthraw(self, wd_amt):
try:
if self.balance >= wd_amt:
self.balance -= wd_amt
print("Withdrwal accepted")
else:
print("Funds unavailable")
except ValueError:
print("valueerror for fund")
```

Q.1 instantiat the class-

```
class account:
```

Q.2 print the object-

```
def deposit(self):
    amount = float(input("Enter amount to be deposited: "))
    self.balance += amount
    print("\n Amount Deposited:", amount)
```

Q.3-

```
def withdraw(self):
    amount = float(input("Enter amount to be withdrawn: "))
    if self.balance >= amount:
```

```
            self.balance -= amount
            print("\n You Withdrew:", amount)
        else:
            print("\n Insufficient balance  ").
```

Q.4-

```
def display(self):
        print("\n Net Available Balance =", self.balance)
```

Q.5-

```
class Bank_Account:
    def __init__(self):
        self.balance=0
        print("Hello!!! Welcome to the Deposit & Withdrawal Machine")

    def deposit(self):
        amount=float(input("Enter amount to be Deposited: "))
        self.balance += amount
        print("\n Amount Deposited:",amount)

    def withdraw(self):
        amount = float(input("Enter amount to be Withdrawn: "))
        if self.balance>=amount:
            self.balance-=amount
            print("\n You Withdrew:", amount)
        else:
            print("\n Insufficient balance  ")

    def display(self):
        print("\n Net Available Balance=",self.balance)
s = Bank_Account()

s.deposit()
```

```
s.withdraw()
s.display()
```

Output:

```
Hello !!! Welcome to Deposit&Withdrawal Machine
Enter amount to be deposited:
 Amount Deposited: 1000.0
Enter amount to be withdrawn:
 You Withdrew: 500.0

 Net Available Balance = 500.0
```