**1. Create a class and overload its greet() method to just print “Hello” if not given any parameters and if parameter is given as "Sansa" then “Hello Sansa”.**  
  
class ovrload{

public static String greet(){

return "Hello";

}

public static String greet(String name){

return "Hello "+ name;

}

public static void main(String[] args){

System.out.println(greet());

System.out.println(greet("sansa"));

}

}

output: 

**2. Create an abstract parent class with abstract methods and create a child class that extends parent class. Override the parent abstract method using child class object.**

abstract class Abrat {

Abrat(){

System.out.println("Abstract Method");

}

public static void print(){

System.out.println("static method");

}

public void print\_1(){

System.out.println("Non- static method");

}

abstract public void Abrat\_method();

}

class Second extends Abrat{

public void Abrat\_method(){

System.out.println("Abstract method has succesfully done");

}

}

class AbratDriver{

public static void main(String[] args) {

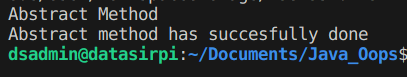
**Abrat aT= new Second();**

**Second sD= (Second)aT;**

sD.Abrat\_method();

}

}

Output:

**3. Add public getter and setter methods to the Store class so its variables can be accessed by . other classes. It should print the store’s name and address and then change both and print the new values**

public class Store {

private String name;

private String address;

Store(String name, String address){

this.name=name;

this.address=address;

}

public String getname(){

return name;

}

public String getaddress(){

return address;

}

public void setname(String name){

this.name=name;

}

public void setaddress(String address){

this.address=address;

}

}

class StoreDriver{

public static void main(String[] args) {

**Store s= new Store("Maanga", "Kilpaakam-627002");**

System.out.println("Old Name :"+s.getname());

System.out.println("Old Address :"+s.getaddress());

s.setname("Theynga");

s.setaddress("meylpaakam-624932");

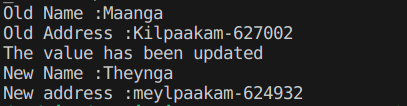
System.out.println("The value has been updated");

System.out.println("New Name :"+s.getname());

System.out.println("New address :"+s.getaddress());

}

}

Output: 

**4. Write a program which implements Multilevel inheritance and check both the child classes can access the methods of super parent class.**

public class Parent {

void details(){

System.out.println("Parent class");

}

}

class Child1 extends Parent{

void print(){

System.out.println("Child\_1");

}

}

class Child2 extends Child1{

void last(){

System.out.println("Child\_2");

}

}

class ParentDrive{

public static void main(String[] args) {

**Child2 c2=new Child2();**

System.out.println("Accessing from child2");

c2.last();

c2.print();

c2.details();

System.out.println("");

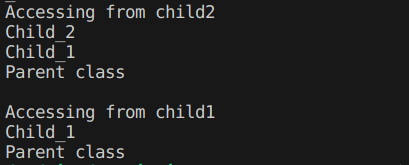
**Child1 c1=new Child1();**

System.out.println("Accessing from child1");

c1.print();

c1.details();

}

}

**5. A class called Account, which models a bank account of a customer, is designed as shown in the following class diagram. The methods credit(amount) and debit(amount) add or subtract the given amount to the balance. The method transferTo(anotherAccount, amount) transfers the given amount from this Account to the given anotherAccount. Write the Account class.**

import java.util.\*;

public class Canara {

private String id;

private String name;

private int balance=0;

Canara(String id, String name) {

this.id=id;

this.name=name;

}

Canara(String id, String name,int balance) {

this.id=id;

this.name=name;

this.balance=balance;

}

public String getID() {

return id;

}

public String getName() {

return name;

}

public int getBalance() {

return balance;

}

public int credit(int amount) {

balance+=amount;

return balance;

}

public int debit(int amount) {

if(balance>=amount) {

balance-=amount;

}

else {

System.out.println("No sufficient balance");

}

return balance;

}

public int transferTo(Canara another, int amount) {

if(balance>=amount) {

balance=balance-amount;

another.balance=another.balance+amount;

System.out.println("Transfer amount to given Account"+ amount);

}

else {

System.out.println("Amount exceeded balance");

}

return balance;

}

public String toString() {

return "Account [id = "+id+" "+"name = "+name+" "+" balance = "+balance+"]";

}

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

Canara bank=new Canara("#12345", "Muthu");

Canara bank2=new Canara("#456718", "Rohan");

System.out.println("Enter amount need to be credited");

int value=sc.nextInt();

System.out.println("Enter amount need to be debit");

int value1=sc.nextInt();

bank.credit(value);

bank.debit(value1);

System.out.println(bank);

System.out.println(bank2.getBalance());

bank.transferTo(bank2,500);

System.out.println("Second Account Details"+bank2);

System.out.println (" Current account1"+bank.getBalance());

System.out.println("Account 2 balance"+bank2.getBalance());

}

}

