PARVATHAM RAM CHARAN

Sampl1.

```
class Student{
   int rollno;    // primitive datatype  + instance variable
   String name;   //reference datatype  + instance variable
   static String org ="CDAC HYD";  //reference datatype  + class varible
   // IF static String org;(not initialized) // we get null

   void Details(int r ,String n){
      rollno=r;
      name=n;
      System.out.println("hello");
   }
    void display(){                    // non-static method
      System.out.println(rollno+" "+name+" "+org);
   }
}
public class Sample1{
   public static void main(String[] args) {
      Student s =  new Student();
      s.display(); // 0 null CDAC HYD
      s.Details(51,"raju");  // 51 raju CDAC HYD
      System.out.println(s.rollno);
      s.display();
      s.Details(10,"vivek");
      s.display();

   }
}
```

// if we wrap data members and methods inside a class , even if they are public or private then this form encapsulation concept;

// after in this encapsulation, if all members are public we can direct access members in outside(within main class also)

//if any thing is private , we cannot access members outside directly

Ouput:

```
PS C:\Users\upendar parvatham\OneDrive\Desktop\Java Class Assigments\OOPS> java Sample1
0 null CDAC HYD
hello
51
51 raju CDAC HYD
hello
10 vivek CDAC HYD
PS C:\Users\upendar parvatham\OneDrive\Desktop\Java Class Assigments\OOPS>
```

Sample2.

```
//No constructor, using a Details() method to set data
class Student{
   int roll;
   String name;
   String Course;
   static String org="C-Dac HYD";
   void display(){
      System.out.println(roll+" "+name+" "+org);
   }
   void Details(int r,String s1 , String s2){
      roll=r;
      name=s1;
      Course=s2;
      display();
   }
```

```
}
public class sample2 {
   public static void main(String[] args) {
     Student s=new Student();


     s.Details(32,"Ram","DAC");
     // 32 Ram C-Dac HYD


   }
}
```

Output:

```
PS C:\Users\upendar parvatham\OneDrive\Desktop\Java Class Assigments\OOPS> java sample2
32 Ram C-Dac HYD
PS C:\Users\upendar parvatham\OneDrive\Desktop\Java Class Assigments\OOPS>
```

Sample3.

```
//Uses a constructor to set data during object creation


class Student{
   int roll;
   int age;
   String name;
   Student(int r,int a,String s){
      roll=r;
      age=a;
      name=s;
   }
   void display(){
      System.out.println(roll+" "+age+" "+name);
   }
}
```

```
public class sample3 {
   public static void main(String[] args) {
      Student s1=new Student(13,22,"Vinay");
      s1.display();
   }
}
```

Output:

```
PS C:\Users\upendar parvatham\OneDrive\Desktop\Java Class Assigments\OOPS> javac sample3.java
PS C:\Users\upendar parvatham\OneDrive\Desktop\Java Class Assigments\OOPS> java sample3
13 22 Vinay
PS C:\Users\upendar parvatham\OneDrive\Desktop\Java Class Assigments\OOPS>
```

Sample 4.

```
 class Person{
   String name ;
   int roll;



 }
public class Sample4 {
   public static void main(String[] args) {
      Person s1 = new Person();
      s1.roll= 10;
      s1.name="ram charan";


      Person s2 = new Person();
      s2.roll= 10;
      s2.name="ram charan";
      System.out.println(s1.roll+"  "+s1.name);
      System.out.println(s2.roll+"  "+s2.name);
   }
}
// if we write this like everything is fine but it lacks code reusabilty and encapsulation;
```

Output:

```
PS C:\Users\upendar parvatham\OneDrive\Desktop\Java Class Assigments\OOPS> javac sample3.java
PS C:\Users\upendar parvatham\OneDrive\Desktop\Java Class Assigments\OOPS> java sample3
13 22 Vinay
PS C:\Users\upendar parvatham\OneDrive\Desktop\Java Class Assigments\OOPS> javac Sample4.java
PS C:\Users\upendar parvatham\OneDrive\Desktop\Java Class Assigments\OOPS> java Sample4
10  ram charan
10  ram charan
PS C:\Users\upendar parvatham\OneDrive\Desktop\Java Class Assigments\OOPS>
```

1.Define a class of type Student that has rollno , age and name as private data members.Define SetData() and GetData() as public member funtions with appropriate functionality.write a program that declares 2 students objects , initializes the first at run time and second by reading from console and then displays both students data.

```
import java.util.*;

class Student{

   private int  rollno;

   private int age;

   private String name;

   public void SetData(int r ,int a , String n){

      rollno= r;

      age=a;

     name = n;

   }

   public void GetData(){

      System.out.println(rollno+" "+age+" "+name);

   }

}

public class Problem1 {
```

```
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);


        Student s1 = new Student();
        s1.SetData(67, 22, "ram");
        //s1.rollno = 67;  // this cant be accsses as roll no is private , if want to access  use given getdata
methods inside(encapsulation)


        Student s2 = new Student();
        int rollno = scanner.nextInt();
        int age = scanner.nextInt();
        String name = scanner.next();
        s2.SetData(rollno, age, name);


        s1.GetData();   //67 22 ram
        s2.GetData();  //60 21 vivek
        scanner.close();


    }
}
```

Output:

```
OUTPUT    DEBUG CONSOLE    PROBLEMS 40    TERMINAL    PORTS

PS C:\Users\upendar parvatham\OneDrive\Desktop\Java Class Assigments\OOPS> javac Problem1.java
PS C:\Users\upendar parvatham\OneDrive\Desktop\Java Class Assigments\OOPS> java Problem1
124
21
raju
67 22 ram
124 21 raju
PS C:\Users\upendar parvatham\OneDrive\Desktop\Java Class Assigments\OOPS>
```

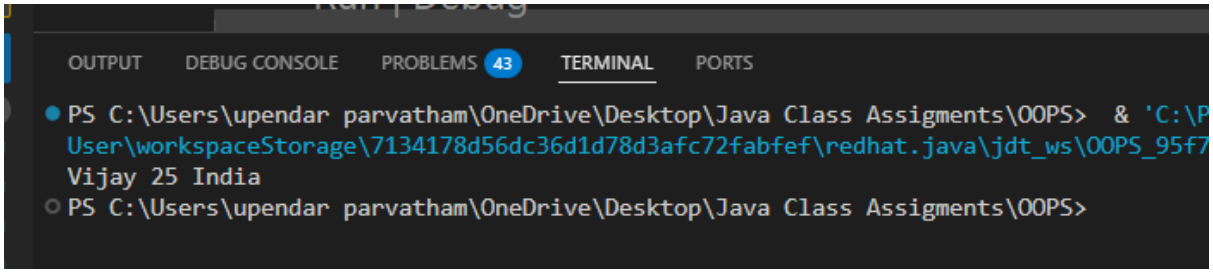
2.Create a class Person with attributes name , age and country .Implement methods to set and get this attributes .Create an object of this class , set its attributes and print out the details.

```
class Person{
    String name;
```

```
    int age;
    String Country;
    Public void set(String n,int a,String c){
        name=n;
        age=a;
        .Country=c;
    }
   Public  void get(){
        System.out.println(name+" "+age+" "+Country+" ");
    }

}
public class Problem2 {
    public static void main(String[] args) {
        Person p=new Person();
        p.set("Vijay",25,"India");
        p.get();
    }
}
```

Output:



3..Constructer Overloading

Extend the person class from the previous problem and add multiple constructors(default,parameterized ,etc...) to initialize

the attributes.Also,include method to display the details.

```
class Person{
   String name;
   int age;
   String Country;
   Person(){      // see last comment line
   }
   Person(String n,int a,String c){
      name=n;
      age =a;
      Country=c;
   }
   Person(String n ,int a){
      name = n;
      age = a;
   }
   void display(){
      System.out.println(name+" "+age+" "+Country);
   }



}
public class Problem3 {
   public static void main(String[] args) {
      Person p1 = new Person();                    //A default constructor
      Person p2 = new Person("ram",21,"India");   //A fully parameterized constructor
      Person p3 = new Person("raju",21);          //A partially parameterized constructor


      p1.display();    //null 0 null
```

```
        p2.display();    //ram 21 India
        p3.display();    //raju 21 null

    }
}
```

//Java only provides a default (no-argument) constructor if you don't write any constructors yourself.

//But as soon as you add any constructor (e.g., the parameterized ones you wrote),

//Java stops generating the default constructor automatically.

Output:

```
OUTPUT    DEBUG CONSOLE    PROBLEMS 42    TERMINAL    PORTS

PS C:\Users\upendar parvatham\OneDrive\Desktop\Java Class Assigments\OOPS> javac Problem3.java
PS C:\Users\upendar parvatham\OneDrive\Desktop\Java Class Assigments\OOPS> java Problem3
null 0 null
ram 21 India
raju 21 null
PS C:\Users\upendar parvatham\OneDrive\Desktop\Java Class Assigments\OOPS>
```

4.Using this : Modify the person class to include a method that displays the name and age of the object .

Use this keyword to differentiate between class variables and method parameters .

Implement a method to campare two Person objects based on their age

```
class Person {
    int age;
    String name;



    void set(int age, String name) {
        this.age = age;
        this.name = name;
    }
```

```
    void display() {
        System.out.println("Name: " + this.name + ", Age: " + this.age);
    }

    // Compare method that compares current object with another Person
    void Compare(Person other) {    //here other is not keyword , we can use any another name also
like
        if (this.age > other.age) {                    //this refers to p1 (the object that called the method)
                                                //other refers to p2 (the object passed in)
            System.out.println(this.name + " is older than " + other.name);
        } else if (this.age < other.age) {
            System.out.println(other.name + " is older than " + this.name);
        } else {
            System.out.println(this.name + " and " + other.name + " are of the same age");
        }
    }
}

public class Problem4 {
    public static void main(String[] args) {
        Person p1 = new Person();
        p1.set(21, "Raju");

        Person p2 = new Person();
        p2.set(29, "Rahul");

        p1.display();
        p2.display();

        // Compare their ages
```

```
        p1.Compare(p2);  // THIS is user defined method
    }
}
```

Output:

```
PS C:\Users\upendar parvatham\OneDrive\Desktop\Java Class Assigments\OOPS> javac Problem4.java
PS C:\Users\upendar parvatham\OneDrive\Desktop\Java Class Assigments\OOPS> java Problem4
Name: Raju, Age: 21
Name: Rahul, Age: 29
Rahul is older than Raju
PS C:\Users\upendar parvatham\OneDrive\Desktop\Java Class Assigments\OOPS>
```

5..Static Variable: Create a class BankAccount with accno,accType,Balance and static variable interestRate.

Initialize it using a static block. Implement methods to deposit and withdraw funds

Create objects and display details.

```
class BankAccount {
    int accno;
    String accType;
    int Balance;
    static int interestRate;



    static {
        interestRate = 5;
    }


    void set(int accno, String accType, int Balance) {
        this.accno = accno;
        this.accType = accType;
        this.Balance = Balance;
    }
```

```
    void get() {
        System.out.println("Account No: " + accno);
        System.out.println("Account Type: " + accType);
        System.out.println("Balance: " + Balance);
        System.out.println("Interest Rate: " + interestRate + "%");
        System.out.println("------------------------------");
    }

    void deposit(int amount) {
        this.Balance += amount;
        System.out.println("Deposited: " + amount);
        System.out.println("New Balance: " + Balance);
        System.out.println("------------------------------");
    }



    void withdraw(int amount) {
        if (amount > Balance) {
            System.out.println("Insufficient balance!");
        } else {
            this.Balance -= amount;
            System.out.println("Withdrawn: " + amount);
            System.out.println("New Balance: " + Balance);
        }
        System.out.println("------------------------------");
    }
}
public class Problem5 {
    public static void main(String[] args) {
```

```
        BankAccount b1 = new BankAccount();
        b1.set(3214, "Savings", 5000);
        b1.get();

        BankAccount b2 = new BankAccount();
        b2.set(2564, "Current", 10000);
        b2.get();

        b1.deposit(1000);
        b2.withdraw(3000);
    }
}
```

Output:

```
PS C:\Users\upendar parvatham\OneDrive\Desktop\Java Class Assigments\OOPS> javac Problem5.java
PS C:\Users\upendar parvatham\OneDrive\Desktop\Java Class Assigments\OOPS> java Problem5
Account No: 3214
Account Type: Savings
Balance: 5000
Interest Rate: 5%
-----------------------------
Account No: 2564
Account Type: Current
Balance: 10000
Interest Rate: 5%
-----------------------------
Deposited: 1000
New Balance: 6000
-----------------------------
Withdrawn: 3000
New Balance: 7000
-----------------------------
PS C:\Users\upendar parvatham\OneDrive\Desktop\Java Class Assigments\OOPS>
```

6.Static Method: Add a static method to the BankAccount class from the previous problem to calculate interest based on a given balance and interest rate. Also, implement a method to display the account details including balance and interest earned.

```
class BankAccount {
  int accno;
  String accType;
  int balance;
  static int interestRate;
```

```
  static {
    interestRate = 5;
  }

  void set(int a, String t, int b) {
    this.accno = a;
    this.accType = t;
    this.balance = b;
  }

  static double calculateInterest(int b) { // here method is static due to method using static varaible
    return (b * interestRate) / 100.0;
  }

  void displayDetails() {
    double interest = calculateInterest(balance);
    System.out.println("Account Number : " + accno);
    System.out.println("Account Type : " + accType);
    System.out.println("Balance : " + balance);
    System.out.println("Interest Rate : " + interestRate + "%");
    System.out.println("Interest Earned : " + interest);

  }

}

public class Problem6 {
  public static void main(String[] args) {
    BankAccount b1 = new BankAccount();
    b1.set(12354, "savings", 2000);
```

```
    BankAccount b2 = new BankAccount();
    b2.set(45678, "Current", 8000);

    b1.displayDetails();
    b2.displayDetails();
  }
}
```

Output:

```
OUTPUT   DEBUG CONSOLE   PROBLEMS 35   TERMINAL   PORTS

PS C:\Users\upendar parvatham\OneDrive\Desktop\Java Class Assigments\OOPS> javac Problem6.java
PS C:\Users\upendar parvatham\OneDrive\Desktop\Java Class Assigments\OOPS> java Problem6
Account Number : 12354
Account Type : savings
Balance : 2000
Interest Rate : 5%
Interest Earned : 100.0
Account Number : 45678
Account Type : Current
Balance : 8000
Interest Rate : 5%
Interest Earned : 400.0
PS C:\Users\upendar parvatham\OneDrive\Desktop\Java Class Assigments\OOPS>
```

7.Using this in Constructors:

Create a class Rectangle with attributes length and width.

Implement a parameterized constructor that initializes these attributes.

Use this to differentiate between class variables and constructor parameters.

Include methods to calculate the area and perimeter.

```
class Rectangle{
   int length;
   int width;
   Rectangle(int l , int w){
      this.length= l;
      this.width = w;
   }
```

```
    void get(){
        System.out.println("length : "+this.length);
        System.out.println("width : "+this.width);
    }
    void calculate(){
        int perimeter = 2*(length+width);
        double area = length*width;
        System.out.println("Perimeter : "+perimeter);
        System.out.println("Area "+area);
    }
}
public class Problem7 {
    public static void main(String[] args) {
        Rectangle r1  = new Rectangle(2, 3);
        r1.get();
        r1.calculate();
    }
}
```

Output:

```
OUTPUT    DEBUG CONSOLE    PROBLEMS 35    TERMINAL    PORTS

PS C:\Users\upendar parvatham\OneDrive\Desktop\Java Class Assigments\OOPS> javac Problem7.java
PS C:\Users\upendar parvatham\OneDrive\Desktop\Java Class Assigments\OOPS> java Problem7
length : 2
width : 3
Perimeter : 10
Area 6.0
PS C:\Users\upendar parvatham\OneDrive\Desktop\Java Class Assigments\OOPS>
```

8.Class and methods:

Create a class Calculator with relevant data members and a constructor.

Implement methods for basic arithmetic operations

(addition, subtraction, multiplication, division, modulus) and demonstrate their usage.

```
class Calculator{
```

```
int  num1;
int num2;
Calculator(){

}
Calculator(int n1 ,int n2){
   this.num1 = n1;
   this.num2 = n2;
}
int add(){
   return num1+num2;
}
int subtract(){
   return num1-num2;
}
int multiply(){
   return num1*num2;
}
int division(){
   if(num2==0){
      System.out.println("Error: Division by zero");
      return 0;
   }
   return num1/num2;
}
int modulus(){
   if (num2 == 0) {
      System.out.println("Error: Modulus by zero!");
      return 0;
   }
   return num1 % num2;
```

```
    }
}
public class Problem8 {
  public static void main(String[] args) {
    Calculator c1 = new Calculator(5,2);


    System.out.println("Addition is : "+c1.add());
    System.out.println("Subtraction is : "+c1.subtract());
    System.out.println("Multiplication is : "+c1.multiply());
    System.out.println("division is : "+c1.division());
    System.out.println("Modulus is : "+c1.modulus());



  }
}
```

Output:

```
PS C:\Users\upendar parvatham\OneDrive\Desktop\Java Class Assigments\OOPS> javac Problem8.java
PS C:\Users\upendar parvatham\OneDrive\Desktop\Java Class Assigments\OOPS> java Problem8
Addition is : 7
Subtraction is : 3
Multiplication is : 10
division is : 2
Modulus is : 1
PS C:\Users\upendar parvatham\OneDrive\Desktop\Java Class Assigments\OOPS>
```

9. //Composition and Aggregation:

//Create a class Address with attributes street, city, and state.

//Then create a class Person with attributes name and an Address object.

//Demonstrate how to use com Write a Java class representing a Student.

//Encapsulate the student's name, age, and grade point average (GPA) with private access modifiers.

//Provide getter and setter methods to access and modify these

```
//attributes position to model the relationship between a person and their address

// Address class
class Address {
   String street;
   String city;
   String state;

   Address(String street, String city, String state) {
      this.street = street;
      this.city = city;
      this.state = state;
   }
}

// Person class
class Person {
   String name;
   Address address;

   Person(String name, Address address) {
      this.name = name;
      this.address = address;
   }
}

// Student class with encapsulation
class Student {
   private String name;
   private int age;
   private double gpa;
```

```
private Person person;

Student(String name, int age, double gpa, Person person) {
    this.name = name;
    this.age = age;
    this.gpa = gpa;
    this.person = person;
}

// Getters
public String getName() {
    return name;
}

public int getAge() {
    return age;
}

public double getGpa() {
    return gpa;
}

public Person getPerson() {
    return person;
}

// Setters
public void setName(String name) {
    this.name = name;
}
```

```
    public void setAge(int age) {
        this.age = age;
    }

    public void setGpa(double gpa) {
        this.gpa = gpa;
    }

    public void setPerson(Person person) {
        this.person = person;
    }

    // Display method
    public void displayStudent() {
        System.out.println("Student Name: " + name);
        System.out.println("Age: " + age);
        System.out.println("GPA: " + gpa);
        System.out.println("Person Name: " + person.name);
        System.out.println("Street: " + person.address.street);
        System.out.println("City: " + person.address.city);
        System.out.println("State: " + person.address.state);
    }
}

// Main class
public class Problem9 {
    public static void main(String[] args) {
        // Create Address object
        Address addr = new Address("MG Road", "Hyderabad", "Telangana");

        // Create Person object
```

```
        Person person = new Person("Ram Charan", addr);

        // Create Student object
        Student student = new Student("Ram Charan", 21, 9.1, person);

        // Display using getters
        System.out.println("Using getters:");
        System.out.println("Name: " + student.getName());
        System.out.println("Age: " + student.getAge());
        System.out.println("GPA: " + student.getGpa());
        System.out.println("Person Name: " + student.getPerson().name);
        System.out.println("Street: " + student.getPerson().address.street);
        System.out.println("City: " + student.getPerson().address.city);
        System.out.println("State: " + student.getPerson().address.state);

        System.out.println("\nModifying values using setters...");

        // Update values using setters
        student.setName("Sivya");
        student.setAge(20);
        student.setGpa(9.5);
        student.getPerson().name = "Sivya"; // Directly updating Person name
        student.getPerson().address.street = "Banjara Hills";
        student.getPerson().address.city = "Hyderabad";
        student.getPerson().address.state = "Telangana";

        // Display updated details
        System.out.println("\nAfter updates:");
        student.displayStudent();
    }
}
```

Output:

```
PS C:\Users\upendar parvatham\OneDrive\Desktop\Java Class Assigments\OOPS> java Problem9
Using getters:
Name: Ram Charan
Age: 21
GPA: 9.1
Person Name: Ram Charan
Street: MG Road
City: Hyderabad
State: Telangana

Modifying values using setters...

After updates:
Student Name: Sivya
Age: 20
GPA: 9.5
Person Name: Sivya
Street: Banjara Hills
City: Hyderabad
State: Telangana
PS C:\Users\upendar parvatham\OneDrive\Desktop\Java Class Assigments\OOPS>
```

10. // write Java program that models a Library.

```
//Create classes for Library, Book, and Author.
// Ensure that the Library class aggregates a collection of Book objects,
// and each Book object has an aggregation relationship with an Author object.
import java.util.List;
import java.util.ArrayList;
class Library{
   String Name;
   String Address;
   List<Book> books; // aggregation of Books
   Library(String N,String A ){
      this.Name = N;
      this.Address=A;
      this.books = new ArrayList<>();  // initialize empty list
   }
   public void addBook(Book b ){
      books.add(b);
```

```
    }
    void displayDetails(){
        System.out.println("Library name : "+ Name);
        System.out.println("Address : "+ Address);
        System.out.println("Books Available:");
        for (Book b : books) {
            System.out.println("  Book Title : " + b.Title);
            System.out.println("  ISBN       : " + b.ISBN);
            System.out.println("  Author     : " + b.author.Name);
            System.out.println("  DOB        : " + b.author.DateofBirth);
            System.out.println("  Nationality: " + b.author.Nationality);
            System.out.println("----------------------------");
        }
    }
}
class Book{
    String Title;
    Author author;
    int ISBN;
    Book(String t , Author a , int n){
        this.Title = t;
        this.author = a;
        this.ISBN= n;
    }
}
class Author{
    String Name;
    String DateofBirth;
    String Nationality;
    Author(String n , String d , String c){
        this.Name = n;
```

```
        this.DateofBirth = d;
        this.Nationality = c;
    }

}

public class Problem10 {
    public static void main(String[] args) {
        Author auth1 = new Author("Ram", "07/11/2002", "Indian");
        Author auth2 = new Author("Charan", "15/05/1999", "Indian");

        Book bk1 = new Book("My Life", auth1, 123456);
        Book bk2 = new Book("Journey of Dreams", auth2, 987654);

        Library lb = new Library("Genius Library", "Hyderabad");

        lb.addBook(bk1);
        lb.addBook(bk2);
        lb.displayDetails();


    }
}
```

Output:

```
PS C:\Users\upendar parvatham\OneDrive\Desktop\Java Class Assigments\OOPS> javac Problem10.java
PS C:\Users\upendar parvatham\OneDrive\Desktop\Java Class Assigments\OOPS> java Problem10
Library name : Genius Library
Address : Hyderabad
Books Available:
  Book Title : My Life
  ISBN       : 123456
  Author     : Ram
  DOB        : 07/11/2002
  Nationality: Indian
---------------------------
  Book Title : Journey of Dreams
  ISBN       : 987654
  Author     : Charan
  DOB        : 15/05/1999
  Nationality: Indian
---------------------------
PS C:\Users\upendar parvatham\OneDrive\Desktop\Java Class Assigments\OOPS>
```