

**(Established under the Presidency University Act, 2013 of the Karnataka Act 41 of 2013)**

**Course Code : CSE1006 Course Title : Problem solving using java**

1. A University wants to develop a an application on student data. The application should allow:
2. Initialise student data using constructor
3. Print student information such as name, roll number and cgpa

class Student {

String name;

int rollNo;

double cgpa;

Student() // default constructor

{

}

Student(String n, int r, double c) { // constructor with parameters

name = n;

rollNo = r;

cgpa = c;

}

// Method to display student details

void display() {

System.out.println("Student Details:");

System.out.println("Name: " + name);

System.out.println("Roll Number: " + rollNo);

System.out.println("CGPA: " + cgpa);

}

public static void main(String[] args) {

Student s1 = new Student("Akhil",111,7.5);

s1.display();

Student s2 = new Student("Kaveri",222,6.4);

s2.display();

}

}

===============

Activity : students can practice the below program which is similar to Student class program

// Demo of employee tax calculation using public methods

import java.util.Scanner;

public class Employee{

  String name;

  int empid;

  double salary,tax;

public void readData(){

  Scanner sc=new Scanner(System.in);

   System.out.println("enter employee name, id and salary");

   name=sc.next();

   empid=sc.nextInt();

   salary=sc.nextFloat();

}

public void calculateTax(){

    if(salary<=50000)

      tax=salary\*5/100;

    else

      tax=salary\*7/100;

}

public void printData()

{

    System.out.println("Employee Details are ");

    System.out.println("name is  "+name );

    System.out.println("employee id is "+empid );

    System.out.println("salary is "+salary );

    System.out.println("tax is "+tax);

}

public static void main(String args[]){

Employee emp1=new Employee();

Employee emp2=new Employee();

emp1.readData();

emp1.calculateTax();

emp2.readData();

emp2.calculateTax();

emp1.printData();

emp2.printData();

}

}

**Test Case :**

enter employee name, id and salary

Vishruth

111

45000

enter employee name, id and salary

Kaveri

222

80000

Employee Details are

name is Vishruth

employee id is 111

salary is 45000.0

tax is 2250.0

Employee Details are

name is Kaveri

employee id is 222

salary is 80000.0

tax is 5600.0

===================

// Demo of constructor overloading for Box class

// Demo of default constructor for Box class

class Box {

    double width;

    double height;

    double depth;

    Box(double side)  // single parameter constructor

     {

        width=side;

        height=side;

        depth=side;

     }

    Box(double w, double  h, double  d)  // three parameter constructor

     {

        width=w;

        height=h;

        depth=d;

     }

   void volume()

   {

      double vol=width \* height \* depth;

    System.out.println("volume is "+vol);

  }

  public static void main(String args[]) {

     Box b1= new Box(5.0); // single parameter constructor is called

     b1.volume();

     Box b2= new Box(2.0,3.0,4.0); // three parameter constructor is called

     b2.volume();

} }

Test Case:

F:\JavaPgms>java Box

volume is 125.0

volume is 24.0

======================

// For the parameterized constructor, we can also give same name to input parameters (same as instance variable names) in that case to access instance variables we use **this** key word (shown below)

Box(double width,double  height, double  depth) // three parameter constructor

     {

        this.width= width;

        this.height= height;

        this.depth= depth;

     }

**Note**: If you omit the keyword ***this*** in the above example, the output would be "0"

this key word in java:

The this keyword in Java is a reference variable that refers to the **current object** of a class.

**Note:** The most common use of the this keyword is to eliminate the confusion between class attributes and parameters with the same name (because a class attribute is shadowed by a method or constructor parameters).

**Example:**

s1.getdata(); // here s1 is current object, inside of getdata() method this references to s1

s2.display();// here s2 is current object, inside of display() method this references to s2

**===================**

**Activity**: Modify the above (Box class) program using **Copy constructor** to apply Constructor overloading.

========================

Problem:

An interviewer asked a job seeker to find the biggest of two integer numbers and also biggest of three integer numbers by applying overloading and static context in the program and finally display the output. Think you as a job seeker and solve the problem by satisfying the requirement given by the interviewer.

class Biggest {

       // Method to find the biggest of two numbers

       static int findBiggest(int a, int b) {

        return (a > b) ? a : b;

    }

    // method to find the biggest of three numbers

    static int findBiggest(int a, int b, int c) {

        return (a > b && a > c) ? a : (b > c) ? b : c;

    }

    public static void main(String[] args) {

        int num1 = 10, num2 = 20, num3 = 35;

    System.out.println("Biggest of " + num1 + " and " + num2 + " is: " +Biggest.findBiggest(num1, num2));

    System.out.println("Biggest of " + num1 + ", " + num2 + " and " + num3 + " is: " + Biggest.findBiggest(num1, num2, num3));

    }

}

Test Case:

F:\JavaPgms>java Biggest

Biggest of 10 and 20 is: 20

Biggest of 10, 20 and 35 is: 35

=============================

// demo of method overloading

import java.util.Scanner;

class Rectangle

{

  double length;  // instance variables or data members or characterstics

  double breadth;

  double area;

void getdata(Double side)  // single parameter

{

       breadth=length=side;

}

void getdata(double length, double breadth) // two parameter method

{

    this.length=length;

    this.breadth=breadth;

}

double area()

{

  return length\*breadth;

}

void display()

{

  System.out.println("length "+length);

  System.out.println("breadth is "+breadth);

  System.out.println("area is "+area);

}

public static void main(String args[])

{

   Rectangle r1=new Rectangle();  // object creation or instantiation

   Rectangle r2=new Rectangle();

   Scanner sc=new Scanner(System.in);

    System.out.println("enter side  ");

    double side=sc.nextDouble();

    r1.getdata(side);

    System.out.println("enter length and breadth of a rectangle");

    double l=sc.nextDouble();

    double b=sc.nextDouble();

    r2.getdata(l,b);

    System.out.println("area of a square is : "+r1.area());

    System.out.println("area of a rectangle is : "+r2.area());

}}

Test Case:

F:\JavaPgms>java Rectangle

enter side

5.1

enter length and breadth of a rectangle

2.1

3.1

area of a square is : 26.009999999999998

area of a rectangle is : 6.510000000000001

====================

**Solve the below Scenario-Based Question:**

1. **Basic Implementation:**  
   A Company keeps track of the total number of employees using a static variable. Ensure that every time a new Employee object is created, the count increases.
2. **Multiple Object Creation:**  
   If you create three instances of the Employee class in the main method, what will be the value of the employee count? Explain how the static variable behaves.
3. **Static Method Access:**  
   Modify the Company Name by including Land mark and city name using static method. How can this method be accessed from the main method without using an object?

public class EmployeePresi

 {

   int empId; // instance variable

   String empName;  // instance variable

   static int empCount=0; // static variable

   static String companyName ="Presidency University";//static variable

   EmployeePresi(int id, String n)  //constructor

    {

     empId = id;

     empName = n;

     empCount++;

     }

   //method to display the values

   void display ()

     {

      System.out.println(empId+"    "+empName+"   "+companyName);

      }

   static void printEmployeeCount()

   {

      System.out.println("employee count is"+empCount);

   }

   static void changeCompanyName()

   {

     companyName="Presidency University, Itgalapur, Bengaluru";

   }

   public static void main(String args[]){

    EmployeePresi e1=new EmployeePresi(111,"Divya");

    EmployeePresi e2=new EmployeePresi(222,"Raghu");

    e1.display(); // calling  normal method using obejct name

    e2.display();

 EmployeePresi.printEmployeeCount(); // calling static method using class name

 EmployeePresi.changeCompanyName(); // updating static variable using static method

    e1.display();

    e2.display();

 }

}

**Test Case:**

F:\JavaPgms>java EmployeePresi

111 Divya Presidency University

222 Raghu Presidency University

employee count is2

111 Divya Presidency University, Itgalapur, Bengaluru

222 Raghu Presidency University, Itgalapur, Bengaluru

===================

**Activity :** Students have to execute the below program and analyse the importance of static block and how static block executes automatically

public class StaticBlock {

    // static block

       static{

          System.out.println("Hello this is a static block");

          }

      public static void main(String args[])

        {

          System.out.println("This is main method");

         } }

================

**Activity :** Students should analyse the logic for below program and develop java code.

**Problem:**

Assume our Presidency University uses a common water tank to supply water to every block. Each time the water level of the tank reduces based on the usage of each block. The facility manger fills the tank thrice with 500 litres each time when request arrives. D block uses 200litres, E Block uses 300 litres and so on. Display the current water level in the tank each time when the blocks are raised their usage and decide when to call the facility manager to fill the tank.

================

**Activity :** Students should analyse the logic for below problem and develop java code.

Develop a java program with two classes an Outer and Inner class. Each of the class have one data member, the inner class has member function which displays the data members. Demonstrate program to access of both inner and outer class members.

==============