CS6308 – JAVA PROGRAMMING

LAB EXPERIMENT – 2

.

NAME: HEMANTH N

REG NO: 2019503519

1. Copy constructor vs user defined constructor, this write a program to build a scanner that verifies the original copy of the object and the duplicate copy of an object using object id.

**CODE:**

import java.util.\*;

class Student

{

int age;

String name;

long aadhar;

Student(int age,String name,long aadhar)

{

System.out.println("THIS IS ORIGINAL CONSTRUCTOR");

this.age=age;

this.name=name;

this.aadhar=aadhar;

}

Student(Student s)

{

System.out.println("THIS IS COPY CONSTRUCTOR");

age=s.age;

name=s.name;

aadhar=s.aadhar;

}

void getstudent()

{

System.out.printf("AGE= %d\nNAME= %s\nAADHAR= %d",age,name,aadhar);

}

}

public class hems\_lab2\_1

{

public static void main(String [] args)

{

Scanner in=new Scanner(System.in);

System.out.println("HEMANTH N | 2019503519");

int age=in.nextInt();

String temp=in.nextLine();

String name=in.nextLine();

long aadhar=in.nextLong();

Student s1=new Student(age,name,aadhar);

System.out.println(s1);

s1.getstudent();

System.out.println();

Student s2=new Student(s1);

System.out.println(s2);

s2.getstudent();

int hs1= System.identityHashCode(s1);

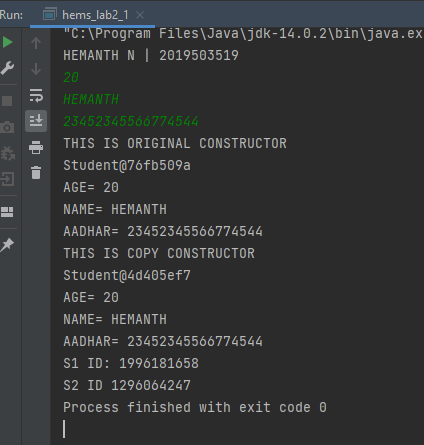
int hs2=System.identityHashCode(s2);

System.out.printf("\nS1 ID: %d\nS2 ID %d",hs1,hs2);

}

}

**OUTPUT:**



// The Id of the objects are different, the contents same. The object id has been verified

1. Calculator-using class, methods and default constructor. Write a program to build a calculator! You are given two real numbers and your task is to print the addition subtraction multiplication division of the two numbers in 4 separate lines. Keep a precision of two digits after decimal

**CODE:**

import java.util.\*;

public class hems\_lab2\_2 {

public static void main(String args[]) {

Scanner in = new Scanner(System.in);

System.out.println("HEMANTH N | 2019503519 \n Enter 2 numbers:");

float a,b;

a = in.nextFloat();

b = in.nextFloat();

calculator calc = new calculator();

calc.calculate(a,b);

}

}

class calculator{

float add(float a, float b) {

return a+b;

}

float sub(float a, float b) {

return a-b;

}

float mul(float a, float b) {

return a\*b;

}

float div(float a, float b) {

return a / b;

}

void calculate(float a, float b) {

System.out.printf("%.2f\n",a+b);

System.out.printf("%.2f\n",a-b);

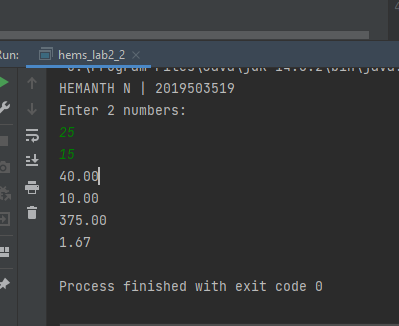
System.out.printf("%.2f\n",a\*b);

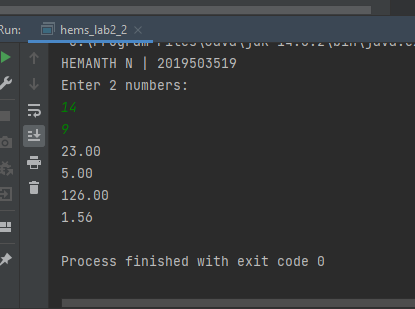
System.out.printf("%.2f\n",a/b);

}

}

**OUTPUT:**





1. Stack and queue. Use container class to execute the required function. Write a program that implements a stackand queue to work within numbers.

**CODE:**

import java.util.Scanner;

public class hems\_lab2\_3 {

public static void main(String args[])

{

Scanner in = new Scanner(System.in);

System.out.println("HEMANTH N | 2019503519");

int n,temp;

System.out.printf("Enter size : ");

n = in.nextInt();

stack s = new stack(n);

queue q = new queue(n);

System.out.println("Enter numbers to be pushed to stack: ");

for(int i=0;i<n;i++) {

temp=in.nextInt();

s.push(temp);

}

System.out.println("Enter numbers to be enqueued to queue: ");

for(int i=0;i<n;i++) {

temp = in.nextInt();

q.enqueue(temp);

}

System.out.println("Stack elements popped are: ");

for(int i=0;i<n;i++)

System.out.printf("%d ", s.pop());

System.out.println("\nStack elements dequeued are: ");

for(int i=0;i<n;i++)

System.out.printf("%d ", q.dequeue());

}

}

class stack {

int st[],top;

void push(int a) {

st[++top] = a;

}

int pop(){

int val = st[top];

top--;

return val;

}

stack(int n) {

st = new int[n];

top = -1;

}

}

class queue {

int q[],front,rear;

void enqueue(int a) {

if(front == -1)

front=0;

q[++rear] = a;

}

int dequeue() {

int val = q[front];

front++;

return val;

}

queue(int n) {

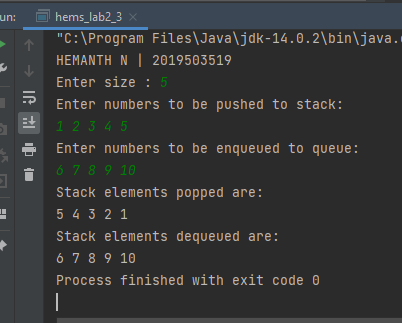
q = new int[n];

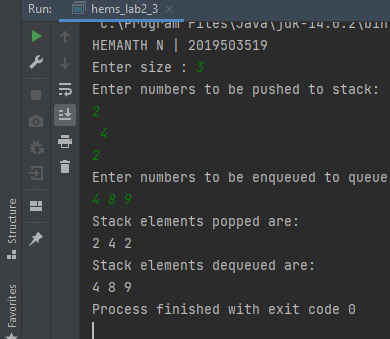
front = rear = -1;

}

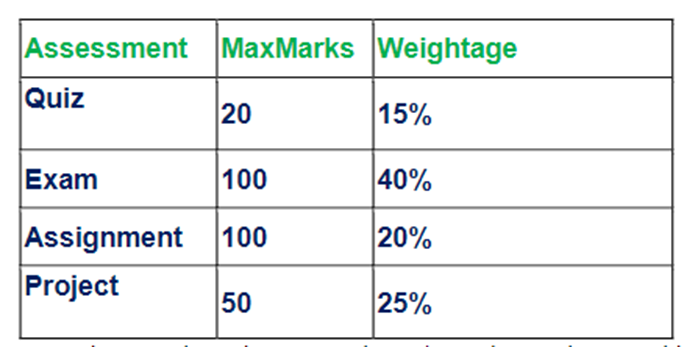
}

**OUTPUT:**





1. Gpa :a course instructor wants to compute the overall gpa of a student based on his performance in the whole semester. The maximum gpa is 10. To evaluate a student, the instructor follows the policy as shown below:



Write a program that initializes the maxmarksand weightage class variables in all four assessments (in that order only) of a studentclass using static blockand based on the policy shown above, the program creates nstudents and the method gpa computes gpa for each student and prints it.now, consider the situation where the instructor makes a mistake while entering the marks. The program has to produce an error message as “error: invalid marks <reason>”. The input is invalid if the marks for any component are either negative (<0) or greater than the corresponding maximum marks. As soon as the program encounters the first invalid entry, an error message with <reason> should be printed.the input consists of 5 lines with first line contains the number of students and following each line containing the marks (non-negative integers) in respective assessment.the output of first line contains the number of studentsand following line containing the gpa of all assessment of each student.

**CODE:**

import java.util.Scanner;

public class hems\_lab2\_4 {

static int[] maxMarks,wetage;

static {

maxMarks = new int[]{20, 100, 100, 50};

wetage = new int[]{15, 40, 20, 25};

}

void gpa(int[][] marks) {

double a,b,c,d,gpa;

for(int i=0;i< marks.length;i++)

{

a = wetage[0]/10.0\*(marks[i][0]/(double)maxMarks[0]);

b = wetage[1]/10.0\*(marks[i][1]/(double)maxMarks[1]);

c = wetage[2]/10.0\*(marks[i][2]/(double)maxMarks[2]);

d = wetage[3]/10.0\*(marks[i][3]/(double)maxMarks[3]);

a = Math.round(a\*100.0)/100.0;

b = Math.round(b\*100.0)/100.0;

c = Math.round(c\*100.0)/100.0;

d = Math.round(d\*100.0)/100.0;

gpa = a+b+c+d;

gpa = Math.round(gpa\*100.0)/100.0;

System.out.println(gpa);

}

}

public static void main(String args[]) {

hems\_lab2\_4 cal = new hems\_lab2\_4();

Scanner in = new Scanner(System.in);

System.out.println("HEMANTH N | 2019503519");

int n,arr[][],temp;

System.out.println("Enter number of students: ");

n = in.nextInt();

arr = new int[n][4];

for(int i=0;i<n;i++) {

for(int j=0;j<4;j++){

temp = in.nextInt();

if(temp<0) {

System.out.printf("ERROR: invalid marks %d<0", temp);

return;

}

else if(temp>maxMarks[j]) {

System.out.printf("ERROR: invalid marks %d > %d", temp, maxMarks[j]);

return;

}

arr[i][j] = temp;

}

}

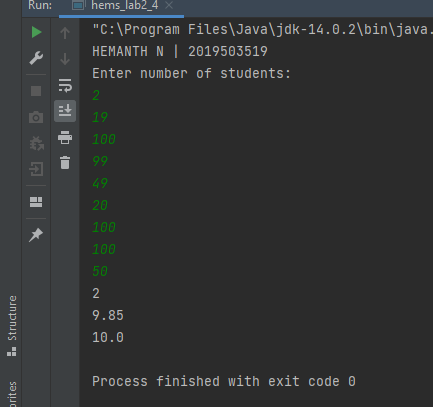
System.out.println(n);

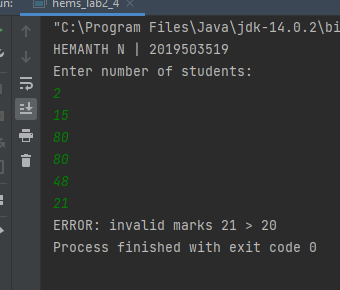
cal.gpa(arr);

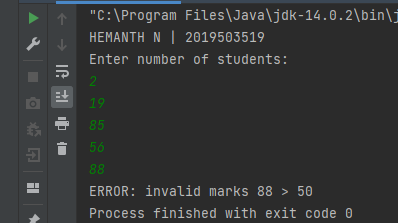
}

}

**OUTPUT:**







\*\*\*\*\*\*