Q1. Write a Program to find the Simple Interest and total amount by

using Command Line Arguments.

Sol.

public class Q1java {

public static void main(String...s) {

int p, r, t, n;

p = Integer.parseInt(s[0]);

r = Integer.parseInt(s[1]);

t = Integer.parseInt(s[2]);

n = Integer.parseInt(s[3]);

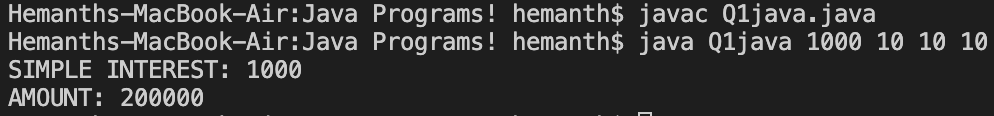
System.out.println("SIMPLE INTEREST: " +

(p \* r \* t / 100) +

"\nAMOUNT: " +

(p\*(1+(r/n))\*(n\*t)));

}

}

Q2. Write a program to find gross salary by using Scanner class. Where basic salary input by the user. hra=10% of basic salary 2, ta=6% of basic salary 3. da=9% of basic salary.

Sol.

import java.util.Scanner;

public class Q2java {

public static void main(String...s) {

Scanner sc = new Scanner(System.in);

int sal;

System.out.print("Enter Your Salary : ");

sal = sc.nextInt();

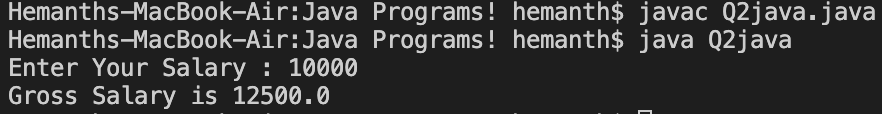
System.out.println("Gross Salary is " +

(sal+(sal\*0.1) +

(sal\*0.06) +

(sal\*0.09)));

}

}

Q3. Write a program to interchange two numbers using a third variable by using Scanner Class.

Sol.

import java.util.\*;

public class Q3java {

public static void main(String...S){

Scanner s= new Scanner(System.in);

int a,b,c;

System.out.print("Enter First number : ");

a=s.nextInt();

System.out.print("Enter Second number : ");

b=s.nextInt();

System.out.println("Before Interchange : a = "+a+" b = "+b);

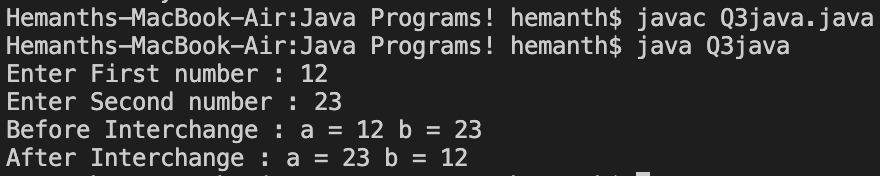
c=a;

a=b;

b=c;

System.out.println("After Interchange : a = "+a+" b = "+b);

}

}

Q4. Write a program to find the lowest and greatest of three numbers in java by using Scanner.

Sol.

import java.util.\*;

public class Q4java {

public static void main(String...S){

Scanner sc = new Scanner(System.in);

int num[] = new int[3];

for(int i = 0; i < 3; i++) {

System.out.print("Enter the num" + (i+1) + ": ");

num[i] = sc.nextInt();

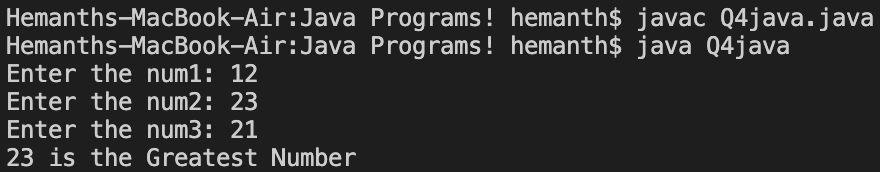
}

for(int i = 0; i < 3; i++) {

num[0] = (num[0] < num[i]) ? num[i] : num[0];

}

System.out.println(num[0] + " is the Greatest Number");

}

}

Q5. Write a program for the addition of two Matrices using Class and Method.

Sol.

import java.util.\*;

class Mat {

Scanner s=new Scanner(System.in);

int[][] mat = new int[2][2];

public void input(int num) {

System.out.println("Enter the matrix" + num + " values: ");

for(int i=0;i<2;i++){

for(int j=0;j<2;j++) {

System.out.print("mat" + num + "[" + i + "][" + j + "]: ");

mat[i][j]=s.nextInt();

}

}

}

Mat(int i) {

input(i);

}

}

public class Q5java {

public static void sum(Mat f,Mat s) {

int[][] to = new int[2][2];

for(int i=0;i<2;i++) {

for(int j=0;j<2;j++) {

to[i][j] = f.mat[i][j] + s.mat[i][j];

}

}

for(int i=0;i<2;i++) {

System.out.print("| ");

for(int j=0;j<2;j++){

System.out.print(to[i][j] + " ");

}

System.out.print("|\n");

}

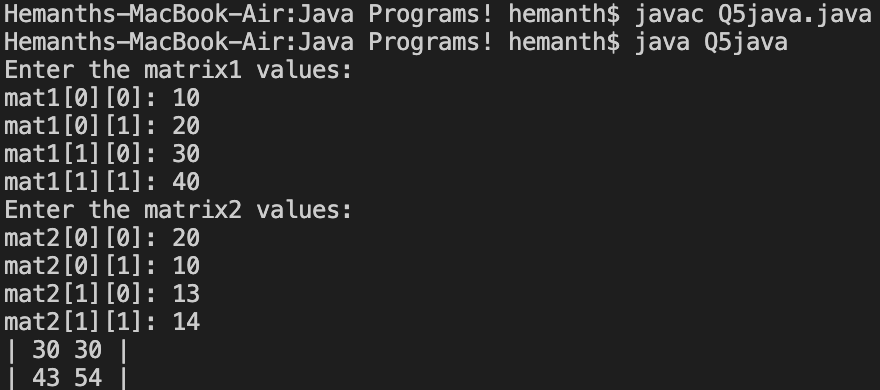
}

public static void main(String...S) {

Mat mat1 = new Mat(1);

Mat mat2 = new Mat(2);

sum(mat1,mat2);

}

}

Q6. Define a class Student to represent student info. Include the following members:

Data members:

a) Sid b) Sname c) Scourse d) Sage

Member Function:

a) input

b) display

c) Show those students whose age &gt; 18. (showage ())

d) Show bca course students (showcourse ())

Sol.

import java.util.\*;

class Student{

int sid, sage;

String sname, scourse;

Scanner sc = new Scanner(System.in);

void input(){

System.out.print("Enter your Details\nID : ");

sid = sc.nextInt();

System.out.print("Name : ");

sname = sc.next();

System.out.print("Age : ");

sage = sc.nextInt();

System.out.print("Course : ");

scourse = sc.next();

}

void display(){

System.out.print("\nID : "+sid+"\nName : "+sname+"\nAge : "+sage+"\nCourse : "+scourse);

}

void showage(){

System.out.print("\n"+(sage>18?(sname+" is "):sname+" is not ")+"above 18.");

}

void showcourse(){

System.out.print("\n"+(scourse.equals("BCA")?"He is a "+scourse+" student.":"He's not a BCA student"));

}

Student() {

input();

}

}

public class Q6java{

public static void main(String...S){

Student s1 = new Student();

Student s2 = new Student();

s1.display();

s1.showage();

s1.showcourse();

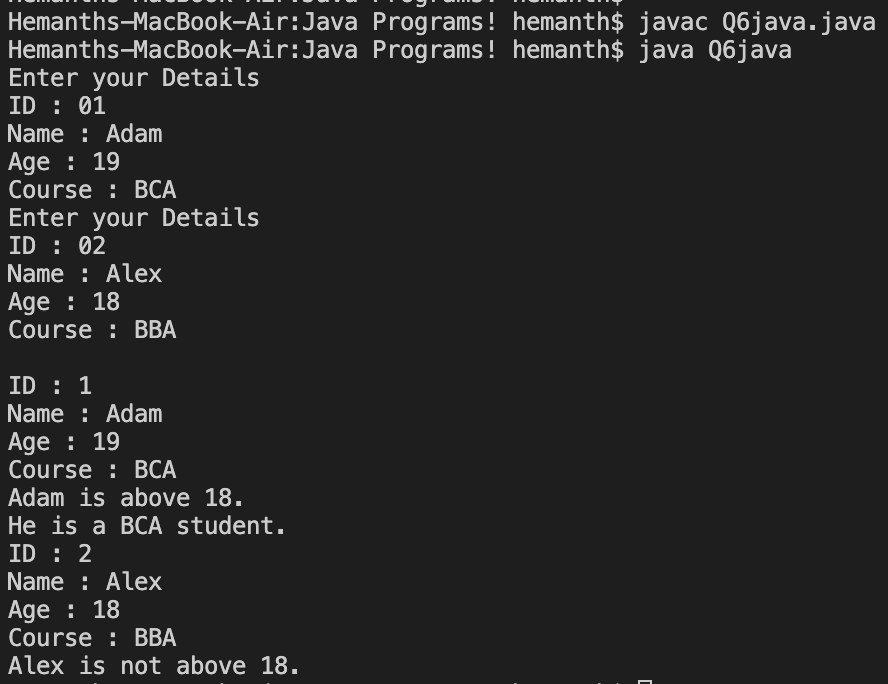
s2.display();

s2.showage();

s2.showcourse();

}

}



Q7. Create a base class teacher. It contains two methods teacherinfo and showinfo for accepting teacher information and display the same. Create a subclass student. It also contains two methods studinfo and showinfo1 for accepting student information and showing the same using single inheritance.

Sol.

import java.util.\*;

class Student{

Scanner sc = new Scanner(System.in);

int sid,sage;

String sname;

public void studinfo(){

System.out.print("Enter the Student's Details : \nID : ");

sid = sc.nextInt();

System.out.print("Name : ");

sname = sc.next();

System.out.print("Age : ");

sage = sc.nextInt();

}

public void showinfo1(){

System.out.println("\nStudent,\n\t\tName : "+sname+"\n\t\tID : "+sid+"\n\t\tAge : "+sage);

}

}

class Teacher extends Student{

int tid,tage;

String tname;

public void teacherinfo(){

System.out.print("Enter the Teacher's Details : \nID : ");

tid = sc.nextInt();

System.out.print("Name : ");

tname = sc.next();

System.out.print("Age : ");

tage = sc.nextInt();

}

public void showinfo(){

System.out.print("\nTeacher,\n\t\tName : "+tname+"\n\t\tID : "+tid+"\n\t\tAge : "+tage);

}

}

public class Q7java {

public static void main(String...S){

Teacher t1 = new Teacher();

t1.teacherinfo();

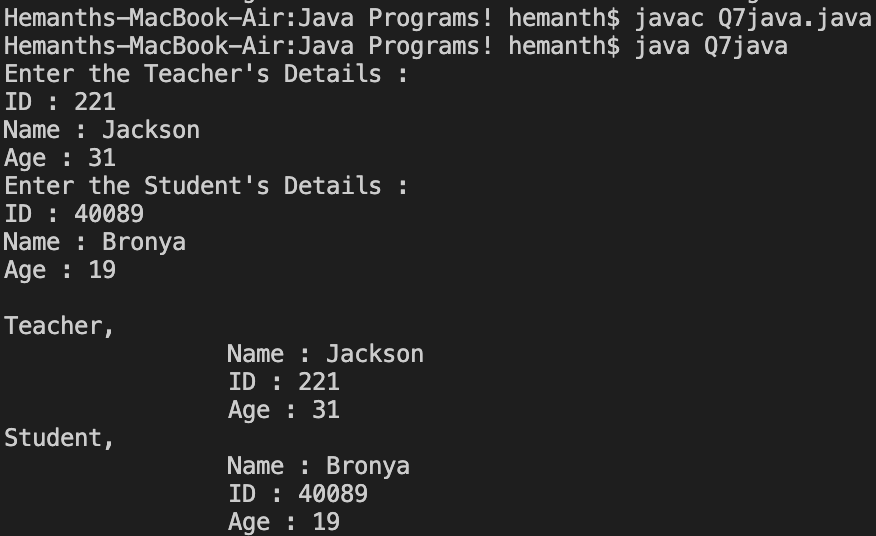
t1.studinfo();

t1.showinfo();

t1.showinfo1();

}

}

Q8. Write a program to use Method Overriding in Java.

Sol.

class College {

public void move() {

System.out.println("College is open!");

}

}

class University extends College {

public void move() {

System.out.println("University is open too!");

}

}

public class Q8java {

public static void main(String args[]) {

College a = new College();

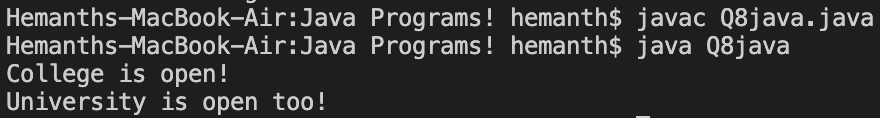
College b = new University();

a.move();

b.move();

}

}



Q9. Write a program to create a class circle to find out the area and circumferences of the circle using this and super keyword.

Sol.

class Circumference {

int r=12;

public void circumference(){

System.out.println("Circumference of Circle is "+(2\*3.14\*this.r));

}

}

class Circle extends Circumference {

public void area(){

System.out.println("Area of Circle is "+(3.14\*super.r\*super.r));

}

}

public class Q9java {

public static void main(String...S){

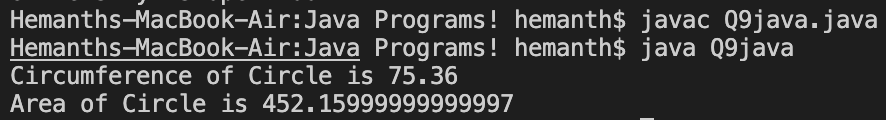
Circumference c1 = new Circumference();

c1.circumference();

Circle c2 = new Circle();

c2.area();

}

}

Q10. Write a program to design a class using abstract method and class.

Sol.

abstract class A {

abstract void m1();

void m2() {

System.out.println("This is inside an Abstract Class.");

}

}

class B extends A {

void m1(){

System.out.println("Implementation of Abstract Method m1.");

}

}

public class Q10java {

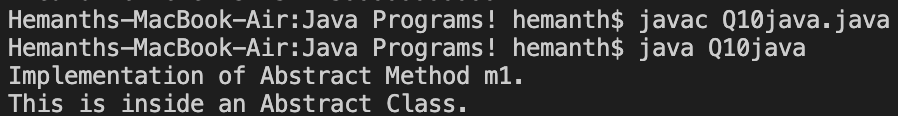
public static void main(String...S){

B b = new B();

b.m1();

b.m2();

}

}

Q11. Write a program to use Method Overriding in Java.

Sol.

public class Q11java {

public static int sum(int x, int y){

return (x + y);

}

public static int sum(int x, int y, int z){

return (x + y + z);

}

public static double sum(double x, double y){

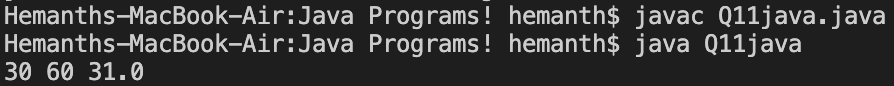
return (x + y);

}

public static void main(String...S){

System.out.println(sum(10, 20)+" "+sum(10, 20, 30)+" "+sum(10.5, 20.5));

}

}

Q12. Write a Program to use Constructor Overloading (Default, Parameterised and Copy Constructor).

Sol.

class Constructors{

Constructors(){

System.out.println("Here I am a \"Default\" constructor.");

}

Constructors(int a){

System.out.println("Parameterized Constructor with value

"+a);

}

Constructors(Constructors c){

System.out.println("Copy Constructor Called.");

}

}

public class Q12java {

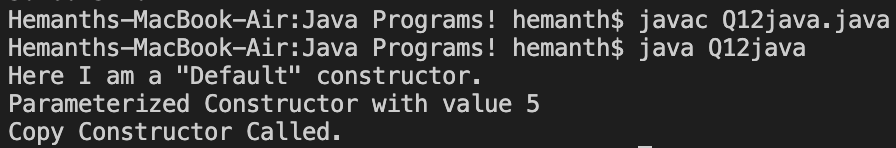
public static void main(String...S){

Constructors c1 = new Constructors();

Constructors c2 = new Constructors(5);

Constructors c3 = new Constructors(c2);

}

}

Q13. Write a Program to use Final Keyword for different types:

1. Final prevents overriding

2. Final prevents inheritance

3. Final works as a constant.

Sol.

final class A { // final class

final void func(){ // final func()

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

}

}

public class Q13java{

static final int pi=3; // constant

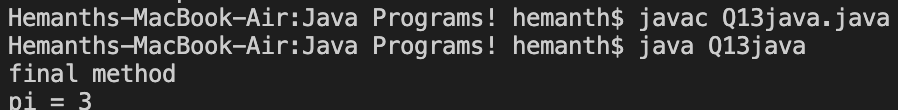
public static void main(String...S){

A a = new A();

a.func();

System.out.println("pi = " + pi);

}

}

Q14. Define two different classes namely, customer and order. These classes are implemented sales interface. The sales interface contains four methods cinput, cshow, oinput and oshow. The customer class has contained cid, cname and cage data. The order class has contained orderid, ordername and orderprice.

Sol.

import java.util.\*;

interface Sales{

public void cinput();

public void cshow();

public void oinput();

public void oshow();

}

class Customer implements Sales{

Scanner sc = new Scanner(System.in);

int cid,cage;

String cname;

public void cinput(){

System.out.print("Enter your Name :");

cname = sc.next();

System.out.print("Enter your ID :");

cid = sc.nextInt();

System.out.print("Enter your Age :");

cage = sc.nextInt();

}

public void cshow(){

System.out.println("Customer's \n\tName : "+cname+

"\n\tID : "+cid+

"\n\tAge : "+cage);

}

public void oshow(){}

public void oinput(){}

}

class Order implements Sales{

Scanner sc = new Scanner(System.in);

int orderid;

float orderprice;

String ordername;

public void oinput(){

System.out.print("Enter Order ID :");

orderid = sc.nextInt();

System.out.print("Enter the Price :");

orderprice = sc.nextFloat();

System.out.print("Enter the Order:");

ordername = sc.next();

}

public void oshow(){

System.out.println("Order's\n\tID : "+orderid+

"\n\tPrice : "+orderprice+

"\n\tName : "+ordername);

}

public void cinput(){}

public void cshow(){}

}

public class Q14java {

public static void main(String[] args) {

Customer C1 = new Customer();

C1.cinput();

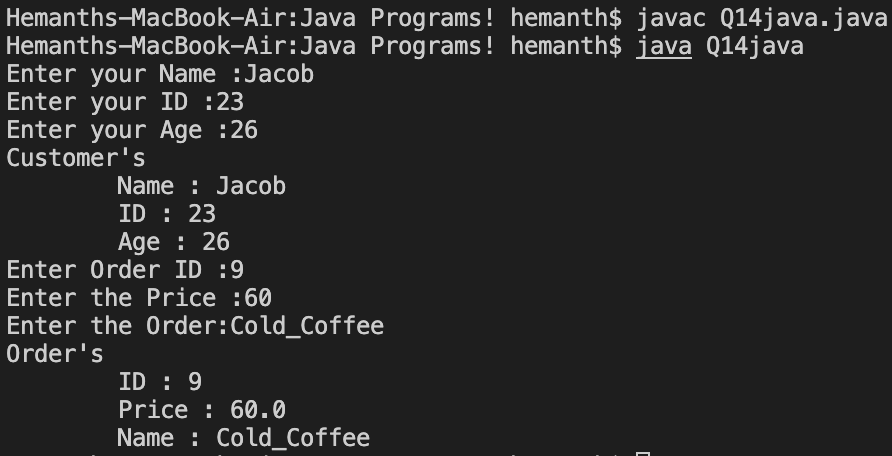
C1.cshow();

Order C2 = new Order();

C2.oinput();

C2.oshow();

}

}

Q15. Write a program to create a package doctor that has contain the doc class with dinfo() and dshow() method after that import the doctor package into another java program that accesses dinfo() and dshow() method from the doc class.

Sol.

package doctor;

import java.util.Scanner;

public class doc{

int did, dage;

String dname;

Scanner sc = new Scanner(System.in);

public void dinfo(){

System.out.print("Enter the Details,\nID : ");

did = sc.nextInt();

System.out.print("Name : ");

dname = sc.next();

System.out.print("Age : ");

dage = sc.nextInt();

}

public void dshow(){

System.out.println("\nID : "+did+"\nName : "+dname+"\nAge : "+dage);

}

}

import doctor.\*;

class Q15java {

public static void main(String...S){

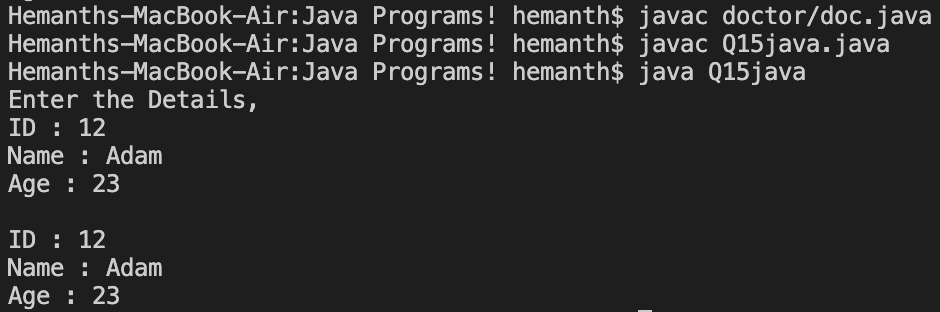
doc d1 = new doc();

d1.dinfo();

d1.dshow();

}

}



Q16. Write a program to create a package employee that has contain emp class with empinfo method. That emp class has contains different field like eid , ename , eage and eaddress. Now import this employee package to another java program and access empinfo method of this file.

Sol.

package employee;

import java.util.\*;

public class emp {

int eid,eage;

String ename, eaddress;

Scanner sc = new Scanner(System.in);

public void empinfo(){

System.out.print("Enter your Details,\nID : ");

eid = sc.nextInt();

System.out.print("Name : ");

ename = sc.next();

System.out.print("Age : ");

eage = sc.nextInt();

System.out.print("Address : ");

eaddress = sc.next();

}

public void eshow() {

System.out.println("\nID : "+eid+"\nName : "+ename+"\nAge : "+eage+"\nAddress : "+eaddress);

}

}

import employee.\*;

public class Q16java {

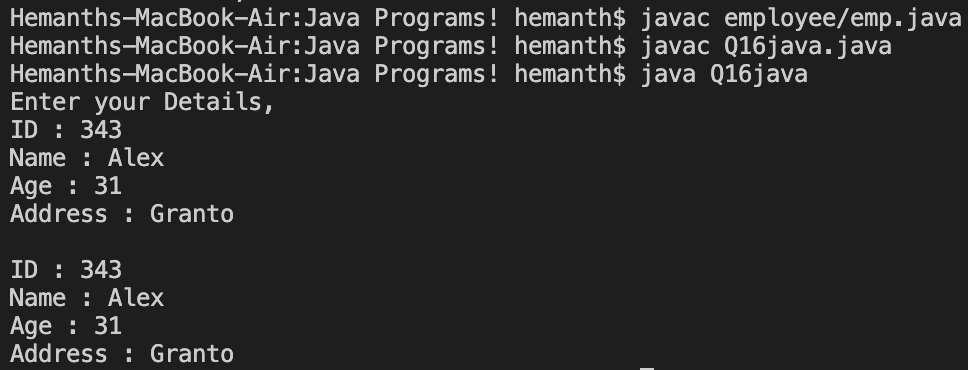
public static void main(String...S){

emp e1 = new emp();

e1.empinfo();

e1.eshow();

}

}

Q17. Write a program to use try, catch, finally and throws by using Exception Handling.

Sol.

import java.io.\*;

public class Q17java {

public static void main(String[] args) throws IOException {

try{

System.out.println("Dividing by Zero Inside try:");

int a=10/0;

}

catch(Exception e){

System.out.println("Got Error : "+e.getMessage());

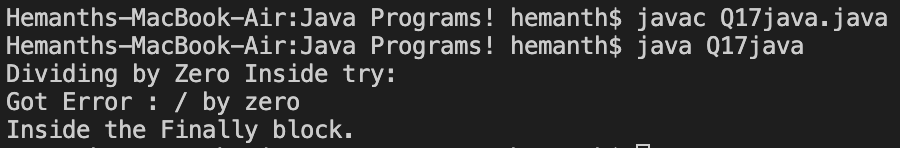
}

finally{

System.out.println("Inside the Finally block.");

}

}

}

Q18. Write a program to use nested try statements.

Sol.

public class Q18java {

public static void main(String[] args) {

try{

try{

System.out.println("Divide by Zero"+(10/0));

} catch(ArithmeticException a){

System.out.println("The exception occurred: "+a.getMessage());

}

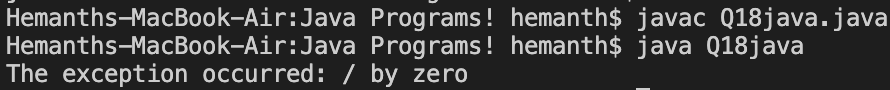
} catch(Exception e){

System.out.println("Handled all Exceptions.");

}

}

}



Q19. Write a program to create a user define Exception using throw keyword.

Sol.

public class Q19java {

static void checkOdd(int num) {

if(num%2 == 0) {

throw new ArithmeticException("Even not allowed!");

}

}

public static void main(String...s) {

try {

checkOdd(14);

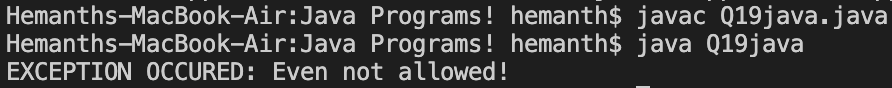
} catch(Exception e) {

System.out.println("EXCEPTION OCCURED: " + e.getMessage());

}

}

}



Q20. Write an application that executes two threads. One thread displays “Hello I am BCA Second Year Student.” every 1000 milliseconds and other displays “I am Well” every 3000 milliseconds. Create these threads by extending the Thread class.

Sol.

class Thread1 extends Thread {

public void run() {

for(int i = 0; i < 3; i++) {

try {

sleep(1000);

System.out.println("Hello I am BCA Second Year Student");

} catch(Exception e) {

System.out.println("Exception caught!");

}

}

}

}

class Thread2 extends Thread {

public void run() {

for(int i = 0; i < 3; i++) {

try {

sleep(3000);

System.out.println("I am Well");

} catch(Exception e) {

System.out.println("Exception caught!");

}

}

}

}

public class Q20java {

public static void main(String...s) {

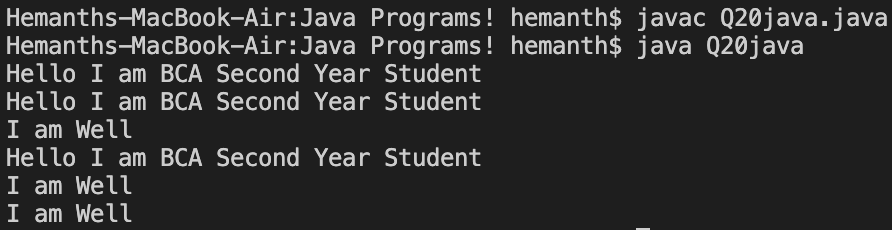
Thread1 t1 = new Thread1();

Thread2 t2 = new Thread2();

t1.start();

t2.start();

}

}

Q21. Write an application that executes two threads. One thread displays “Hello I am BCA Second Year Student.” every 1000 milliseconds and other displays “I am Well” every 3000 milliseconds. Create these threads by implements using runnable interface.

Sol.

class Thread1 implements Runnable {

public void run() {

for(int i = 0; i < 3; i++) {

try {

Thread.currentThread().sleep(1000);

System.out.println("Hello I am BCA Second Year Student");

} catch(Exception e) {

System.out.println("Exception caught!");

}

}

}

}

class Thread2 implements Runnable {

public void run() {

for(int i = 0; i < 3; i++) {

try {

Thread.currentThread().sleep(3000);

System.out.println("I am Well");

} catch(Exception e) {

System.out.println("Exception caught!");

}

}

}

}

public class Q21java {

public static void main(String...s) {

Thread th1 = new Thread(new Thread1());

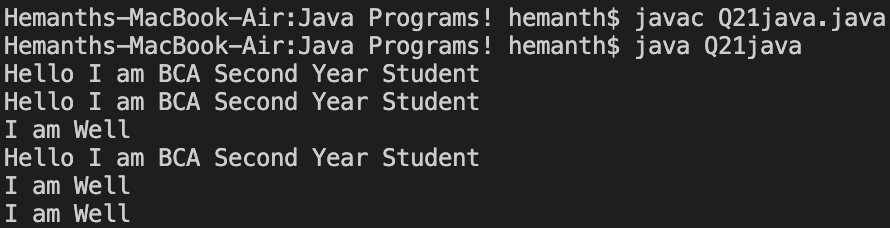
Thread th2 = new Thread(new Thread2());

th1.start();

th2.start();

}

}



Q22. Write a program to print table by using synchronization in java.

Sol.

class Table {

void display(int num) {

synchronized(this) {

for(int i = 1; i <= 10; i++) {

System.out.println(num + " x " + i + " = " + (num \* i));

try {

Thread.sleep(500);

} catch(Exception e) {

System.out.println("Exception caught: \"" + e.getMessage() + "\"");

}

}

}

}

}

class Thread1 extends Thread {

Table t;

Thread1(Table t) {

this.t = t;

}

public void run() {

t.display(5);

}

}

public class Q22java {

public static void main(String...s) {

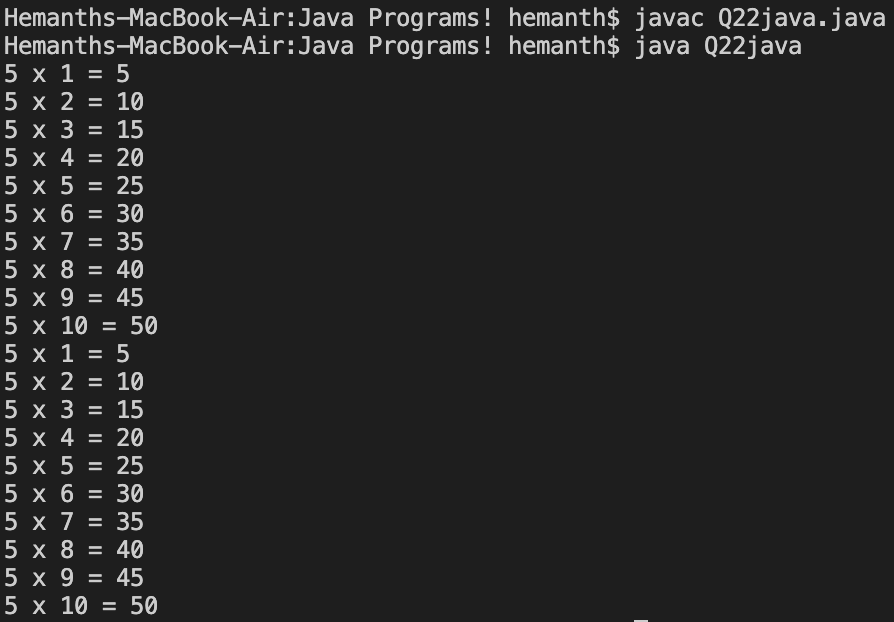
Table t = new Table();

Thread1 t1 = new Thread1(t);

Thread1 t2 = new Thread1(t);

t1.start();

t2.start();

}

}

Q23. Write a program to use these String class functions with suitable example:

* 1. charAt() 2. length () 3. equals() 4. trim() 5. substring()

Sol.

public class Q23java {

public static void main(String...S) {

String s = " Hemanth ";

System.out.println("s.charAt(2): " + s.charAt(2));

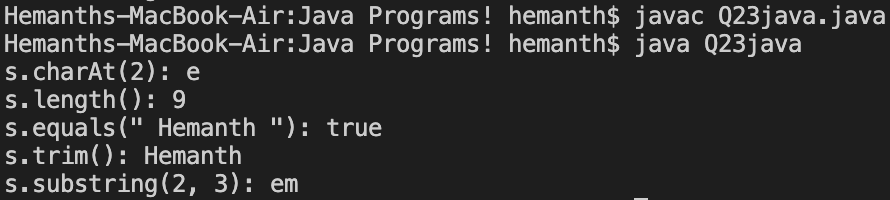
System.out.println("s.length(): " + s.length());

System.out.println("s.equals(\" Hemanth \"): " + s.equals(" Hemanth "));

System.out.println("s.trim(): " + s.trim());

System.out.println("s.substring(2, 3): " + s.substring(2, 4));

}

}

Q24. Write a program to use these String Buffer class functions with a suitable example:

1. append () 2. capacity () 3. delete () 4. insert ()

Sol.

public class Q24java {

public static void main(String...S) {

StringBuffer s = new StringBuffer("Hemanth");

System.out.println("s.append(\" Singh\"): " + s.append(" Singh"));

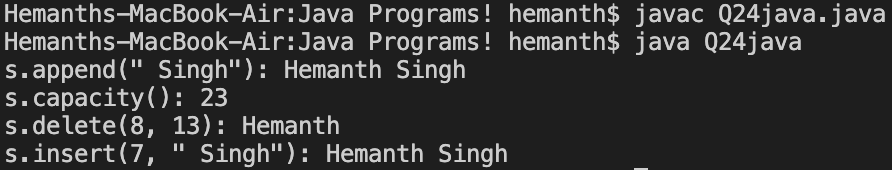
System.out.println("s.capacity(): " + s.capacity());

System.out.println("s.delete(8, 13): " + s.delete(8, 13));

System.out.println("s.insert(7, \" Singh\"): " + s.insert(7, " Singh"));

}

}



Q25. Write a program to store doctor information in a file and display the same. Doctor information is: did, dname, daddress and dage.

Sol.

import java.io.\*;

public class Q25java {

public static void writing(FileWriter ff, String s) throws IOException {

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

ff.write(s.charAt(i));

}

ff.write(" ");

}

public static void main(String[] args) throws IOException {

BufferedReader b = new BufferedReader(new InputStreamReader(System.in));

String dname, did, dage, dadd;

int r;

System.out.print("Enter the Details :\n ID : ");

did = b.readLine();

System.out.print("Name :");

dname = b.readLine();

System.out.print("Age :");

dage = b.readLine();

System.out.print("Address :");

dadd = b.readLine();

FileWriter fi = new FileWriter("doc.txt");

writing(fi, did);

writing(fi, dname);

writing(fi, dage);

writing(fi, dadd);

fi.close();

System.out.println("Reading From the file");

FileReader fr = new FileReader("doc.txt");

while ((r = fr.read()) != -1) {

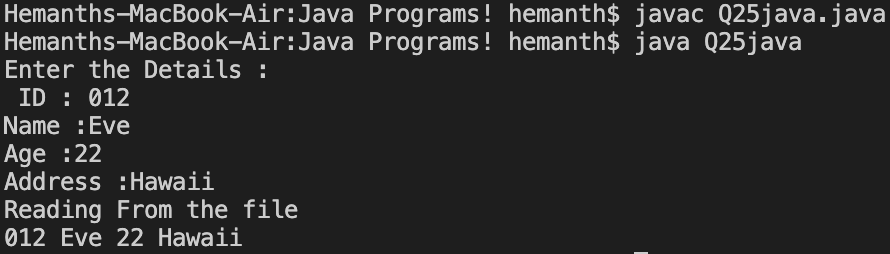
System.out.print((char) r);

}

System.out.println();

fr.close();

}

}

Q26. Write a program to store teacher and student information in a file and display the same. Teacher information is: tid , tname , tage and student information is : sid , sname , sage and saddress.

Sol.

import java.io.\*;

public class Q26java {

public static void writing(FileWriter ff, String s) throws IOException {

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

ff.write(s.charAt(i));

}

ff.write(" ");

}

public static void main(String[] args) throws IOException {

BufferedReader b = new BufferedReader(new InputStreamReader(System.in));

String sname,tname, sid,tid, sage,tage, sadd,tadd;

int r;

System.out.print("Enter the Details of Student : \nID : ");

sid = b.readLine();

System.out.print("Name :");

sname = b.readLine();

System.out.print("Age :");

sage = b.readLine();

System.out.print("Address :");

sadd = b.readLine();

System.out.print("Enter the Details of Teacher : \nID : ");

tid = b.readLine();

System.out.print("Name :");

tname = b.readLine();

System.out.print("Age :");

tage = b.readLine();

System.out.print("Address :");

tadd = b.readLine();

FileWriter fi = new FileWriter("doct.txt");

writing(fi, sid);

writing(fi, sname);

writing(fi, sage);

writing(fi, sadd);

fi.write("\n");

writing(fi, tid);

writing(fi, tname);

writing(fi, tage);

writing(fi, tadd);

fi.close();

System.out.println("Reading From the file");

FileReader fr = new FileReader("doct.txt");

while ((r = fr.read()) != -1) {

System.out.print((char) r);

}

System.out.println();

fr.close();

}

}

