DRONACHARYA
College of Engineering

LABORATORY MANUAL

B.Tech. Semester- IV

WEB TECHNOLOGIES LAB
Subject code: LC-CSE-216G

Prepared by:

Prof. Anita

Sign.: …………………….

Checked by:

Prof. Yashwardhan Soni

Sign.: …………………..

Approved by:

Name : Prof. (Dr.) Isha Malhotra

Sign.: …………………

DEPARTMENT OF CSE/CSIT/IT/IOT
DRONACHARYA COLLEGE OF ENGINEERING
KHENTAWAS, FARRUKH NAGAR, GURUGRAM (HARYANA)

Table of Contents

Vision and Mission of the Institute

Vision:

"Empowering human values and advanced technical education to navigate and address global challenges with excellence."

Mission:

- **M1:** Seamlessly integrate human values with advanced technical education.

- **M2:** Supporting the cultivation of a new generation of innovators who are not only skilled but also ethically responsible.

- **M3:** Inspire global citizens who are equipped to create positive and sustainable impact, driving progress towards a more inclusive and harmonious world.

Vision and Mission of the Department

Vision:

"Steering the future of computer science through innovative advancements, fostering ethical

values and principles through technical education."

Mission:

M1: Directing future innovations in computer science through revolutionary progress.

M2: Instilling a foundation of ethical values and principles in every technologist.

M3: Offering a comprehensive technical education to equip individuals for a meaningful

and influential future.

Programme Educational Objectives (PEOs)

PEO1: Apply the technical competence in Computer Science and Engineering for solving problems in the real world.

PEO2: Carry out research and develop solutions on problems of social applications.

PEO3: Work in a corporate environment, demonstrating team skills, work morals, flexibility and lifelong learning.

Programme Outcomes (POs)

PO1: Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO2: Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO3: Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

PO4: Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5: Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

PO6: The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO7: Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO8: Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO9: Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO10: Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO11: Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO12: Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Program Specific Outcomes (PSOs)

PSO1: Exhibit design and programming skills to develop and mechanize business solutions using revolutionary technologies.

PSO2: Learn strong theoretical foundation leading to brilliance and enthusiasm towards research, to provide well-designed solutions to complicated problems.

PSO3: Work effectively with diverse Engineering fields as a team to design, build and develop system applications.

University Syllabus

1. Program to calculate area using class and object.
2. Program to take input from command line
3. Program to take the input data from user using class BufferedReader.
4. Program in JAVA using constructor overloading to calculate volume.
5. Program in JAVA to calculate volume using single inheritance.
6. Program in JAVA to implement multiple inheritance using Interface
7. Program to create and import a package to calculate marks and print the grade of student.
8. Program to set the priority of a thread in Multithreading.
9. Program for handling uncaught exception using finally. 10.Program to show a face on an Applet.

   10. Design and implement a simple shopping cart example with session tracking API.

Course Outcomes (COs)

Upon successful completion of the course, the students will be able to:

C216.1: Develop static web pages using HTML

C216.2: Develop Java programs for window/web-based applications.

C216.3: Design dynamic web pages using JavaScript and XML.

C216.4: Design dynamic web pages using server-side programming e.g. Servlet/JSP.

C216.5: Design server-side applications using JDBC, ODBC and session tracking API.

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
C216.1	1		1		3			2	3	1	3	1
C216.2	1		3		3			2	3	1	3	1
C216.3	1		3		3			2	3	1	3	1
C216.4	1		3		3			2	3	1	3	1
C216.5	1		1		3			2	3	1	3	1
C216	**1**		**2.2**		**3**			**2**	**3**	**1**	**3**	**1**

CO-PSO Mapping

	PSO1	PSO2	PSO3
C216.1	2		2
C216.2	2		2
C216.3	2		2
C216.4	2		2
C216.5	2		2
C216	**2**		**2**

Course Overview

In order to make websites look and function a certain way, web developers utilize different languages. The three core languages that make up the World Wide Web are HTML5, CSS, and JavaScript. In the IT world, the internet is an essential platform, whether its for developing or for consumer use. When developing a website, typically three main languages come into play. These languages are JavaScript, CSS, and HTML. HTML is the backbone of most webpages. Essentially, it is used to create the structure of how a specific website would look like, from the headings, to the paragraphs, the body, links, and even images.

This course is intended to teach the basics involved in publishing content on the World Wide Web. This includes the 'language of the Web' – HTML, the fundamentals of how the Internet and the Web function, a basic understanding of graphic production with a specific stress on creating graphics for the Web, and a general grounding introduction to more advanced topics such as programming and scripting. This will also expose students to the basic tools and applications used in Web publishing. The objective of this lab is to develop an ability to design and implement static and dynamic website. The courses contain web basics: Design web pages through coding using HTML and DHTML, Browser side scripting using JavaScript with a focus on, event handling and validation, server-side scripting: Php syntax, variables, loops and constructs. Java graphics, Browser side scripting: Introduction to programming world of xml technologies. Basic xml tags, database handling with php and xml, connecting to databases using php, and jdbc.

List of Experiments mapped with COs

Si No.	Name of the Experiment	Course Outcome
1	Program to calculate area using class and object.	C216.1, C216.3
2	Program to take input from command line	C216.3
3	Program to take the input data from user using class BufferedReader.	C216.4
4	Program in JAVA using constructor overloading to calculate volume.	C216.3
5	Program in JAVA to calculate volume using single inheritance.	C216.2
6	Program in JAVA to implement multiple inheritance using Interface	C216.3
7	Program to create and import a package to calculate marks and print the grade of student.	C216.5
8	Program to set the priority of a thread in Multithreading.	C216.4
9	Program for handling uncaught exception using finally.	C216.4
10	Program to show a face on an Applet	C216.5

DOs and DON'Ts

DOs

1. Login-on with your username and password.
2. Log off the computer every time when you leave the Lab.
3. Arrange your chair properly when you are leaving the lab.
4. Put your bags in the designated area.
5. Ask permission to print.

DON'Ts

1. Do not share your username and password.
2. Do not remove or disconnect cables or hardware parts.
3. Do not personalize the computer setting.
4. Do not run programs that continue to execute after you log off.
5. Do not download or install any programs, games or music on computer in Lab.
6. Personal Internet use chat room for Instant Messaging (IM) and Sites is strictly prohibited.
7. No Internet gaming activities allowed.
8. Tea, Coffee, Water & Eatables are not allowed in the Computer Lab.

General Safety Precautions

Precautions (In case of Injury or Electric Shock)

1. To break the victim with live electric source, use an insulator such as fire wood or plastic to break the contact. Do not touch the victim with bare hands to avoid the risk of electrifying yourself.

2. Unplug the risk of faulty equipment. If main circuit breaker is accessible, turn the circuit off.

3. If the victim is unconscious, start resuscitation immediately, use your hands to press the chest in and out to continue breathing function. Use mouth-to-mouth resuscitation if necessary.

4. Immediately call medical emergency and security. Remember! Time is critical; be best.

Precautions (In case of Fire)

1. Turn the equipment off. If power switch is not immediately accessible, take plug off.

2. If fire continues, try to curb the fire, if possible, by using the fire extinguisher or by covering it with a heavy cloth if possible isolate the burning equipment from the other surrounding equipment.

3. Sound the fire alarm by activating the nearest alarm switch located in the hallway.

4. Call security and emergency department immediately:

   Emergency   :   Reception

   Security   :   Front Gate

Guidelines to students for report preparation

All students are required to maintain a record of the experiments conducted by them. Guidelines for its preparation are as follows: -

1) All files must contain a title page followed by an index page. ***The files will not be signed by the faculty without an entry in the index page.***

2) Student's Name, Roll number and date of conduction of experiment must be written on all pages.

3) For each experiment, the record must contain the following

      (i) Aim/Objective of the experiment

      (ii) Pre-experiment work (as given by the faculty)

      (iii) Lab assignment questions and their solutions

      (iv) Test Cases (if applicable to the course)

      (v) Results/ output

Note:

1. Students must bring their lab record along with them whenever they come for the lab.

2. Students must ensure that their lab record is regularly evaluated.

Lab Assessment Criteria

An estimated 10 lab classes are conducted in a semester for each lab course. These lab classes are assessed continuously. Each lab experiment is evaluated based on 5 assessment criteria as shown in following table. Assessed performance in each experiment is used to compute CO attainment aswell as internal marks in the lab course.

Grading Criteria	Exemplary (4)	Competent (3)	Needs Improvement (2)	Poor (1)
AC1: **Pre-Lab written work (this may be assessed through viva)**	Complete procedure with underlined concept is properly written	Underlined concept is written but procedureis incomplete	Not able to write concept and procedure	Underlined concept is not clearly understood
AC2: **Program Writing/ Modeling**	Assigned problem is properly analyzed, correct solution designed, appropriate language constructs/ tools are applied, Program/solution written is readable	Assigned problem is properly analyzed, correct solution designed, appropriate language constructs/ tools are applied	Assigned problem is properly analyzed & correct solution designed	Assigned problem is properly analyzed
AC3: **Identification & Removal of errors/ bugs**	Able to identify errors/ bugs and remove them	Able to identify errors/ bugs and remove them with little bit of guidance	Is dependent totally on someone for identification of errors/ bugs and their removal	Unable to understand the reason for errors/ bugs even after they are explicitly pointed out
AC4:Execution & Demonstration	All variants of input /output are tested, Solution is well demonstrated and implemented concept is clearly explained	All variants of input /output are not tested, However, solution is well demonstrated and implemented concept is clearly explained	Only few variants of input /output are tested, Solution is well demonstrated but implemented concept is not clearly explained	Solution is not well demonstrated and implemented concept is not clearly explained

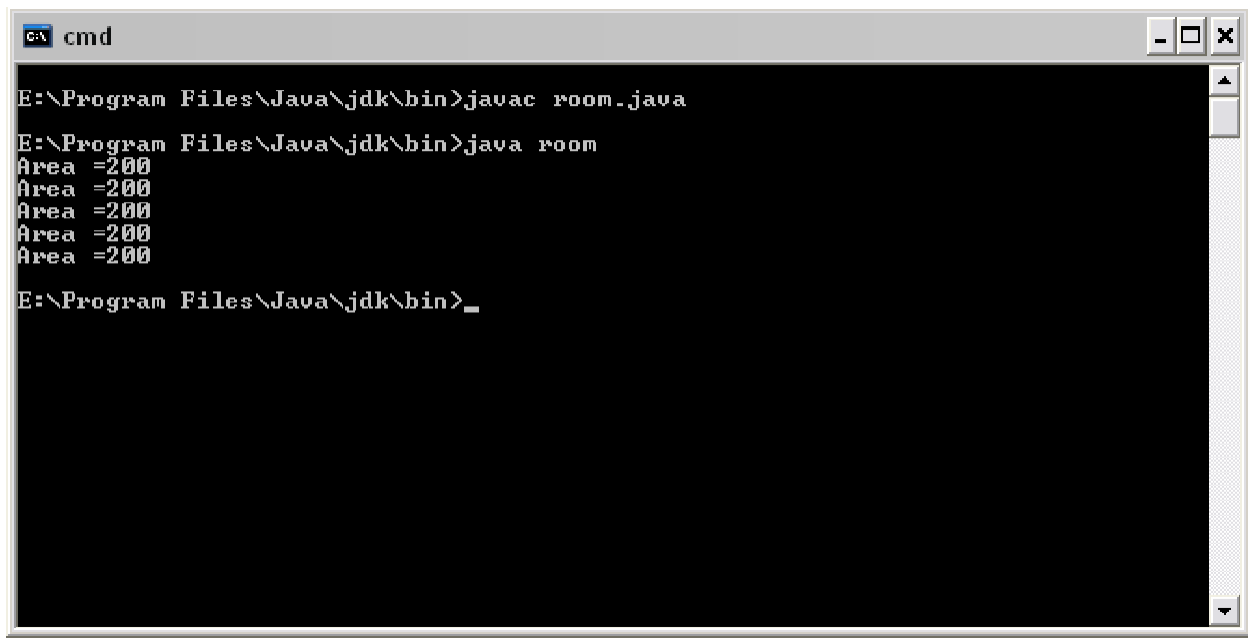|				
AC5:Lab Record Assessment	All assigned problems are well recorded with objective, design constructs and solution along with Performance analysis using all variants of input and output	More than 70 % of the assigned problems are well recorded with objective, design contracts and solution along with Performance analysis is done with all variants of input and output	Less than 70 % of the assigned problems are well recorded with objective, design contracts and solution along with Performance analysis is done with all variants of input and output	Less than 40 % of the assigned problems are well recorded with objective, design contracts and solution along with Performance analysis is done with all variants of input and output

LAB EXPERIMENTS

PROGRAM NO - 1

➢ Write a program to calculate area using class and object.

```
classroomarea

{

intlength,breadth,area;

voidgetdata(intl,int b)

{

length=l;

breadth=

b;

}

intcalarea()

{

area=length*breadth;

System.out.println("Area ="+area);

return area;

}

}
class room

{

public static void main(String args[])

{

roomarea r=new roomarea();

r.getdata(10,20);
```

```
r.calarea();

System.out.println("Area ="+r.calarea());

intaa=r.calarea(); System.out.println("Area

="+aa);

}

}
```

OUTPUT
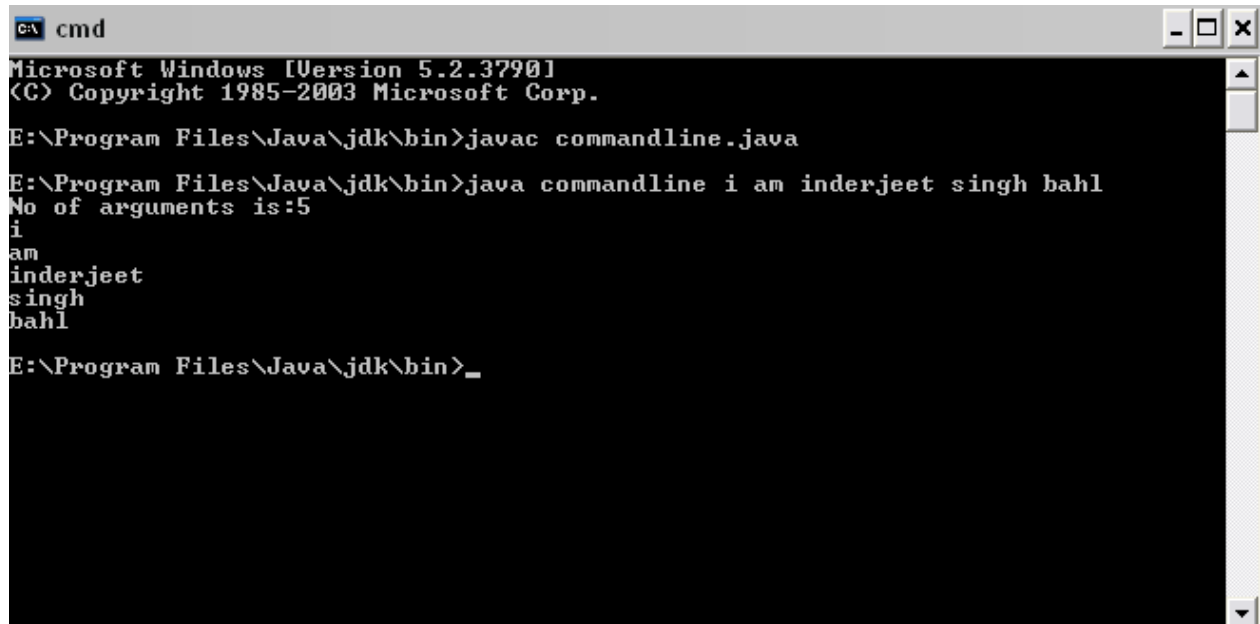
PROGRAM NO - 2

➢ Write a program to take input from command line.

```
classcommandline

{

public static void main(String args[])

{

intcount,i=0;

String st;

count=args.length;

System.out.println("No of arguments:"+count);

while(i<count)

{

st=args[i

];i=i+1;

System.out.println(st);

}

}

}
```

OUTPUT

```
cmd                                                    _ □ ×
Microsoft Windows [Version 5.2.3790]
(C) Copyright 1985-2003 Microsoft Corp.

E:\Program Files\Java\jdk\bin>javac commandline.java

E:\Program Files\Java\jdk\bin>java commandline i am inderjeet singh bahl
No of arguments is:5
i
am
inderjeet
singh
bahl

E:\Program Files\Java\jdk\bin>_
```
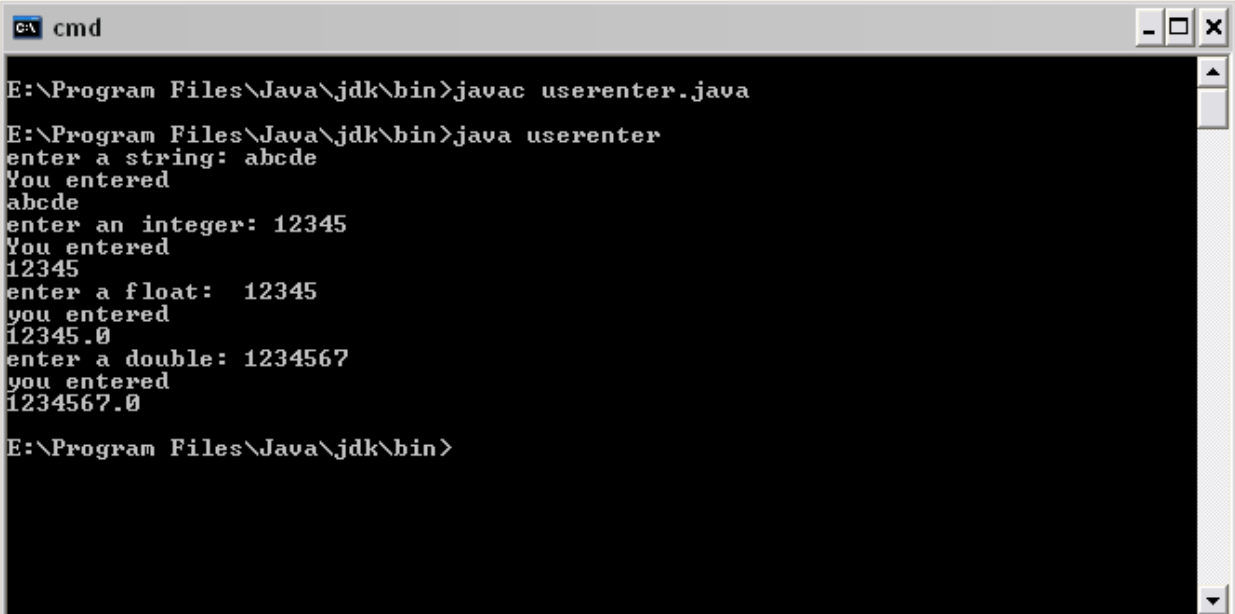
PROGRAM NO – 3

➢ Write a program to take the input data from user using classbufferedReader.

```java
import java.io.*;
classuserenter
{
public static void main(String args[])throws IOException
{
BufferedReaderbr=new BufferedReader(new

InputStreamReader(System.in));
System.out.print("enter a string:\t"); String
str=br.readLine(); System.out.println("You
entered"); System.out.println(str);
System.out.print("enter an integer: ");
inti=Integer.parseInt(br.readLine());
System.out.println("You entered");
System.out.println(i); System.out.print("enter
a float:\t"); float
f=Float.parseFloat(br.readLine());
System.out.println("you entered");
System.out.println(f); System.out.print("enter
a double:\t");
double d=Double.parseDouble(br.readLine());
System.out.println("you entered");
System.out.println(d);
}
}
```

OUTPUT

```
cmd                                                           _ □ ✕

E:\Program Files\Java\jdk\bin>javac userenter.java

E:\Program Files\Java\jdk\bin>java userenter
enter a string: abcde
You entered
abcde
enter an integer: 12345
You entered
12345
enter a float:  12345
you entered
12345.0
enter a double: 1234567
you entered
1234567.0

E:\Program Files\Java\jdk\bin>
```
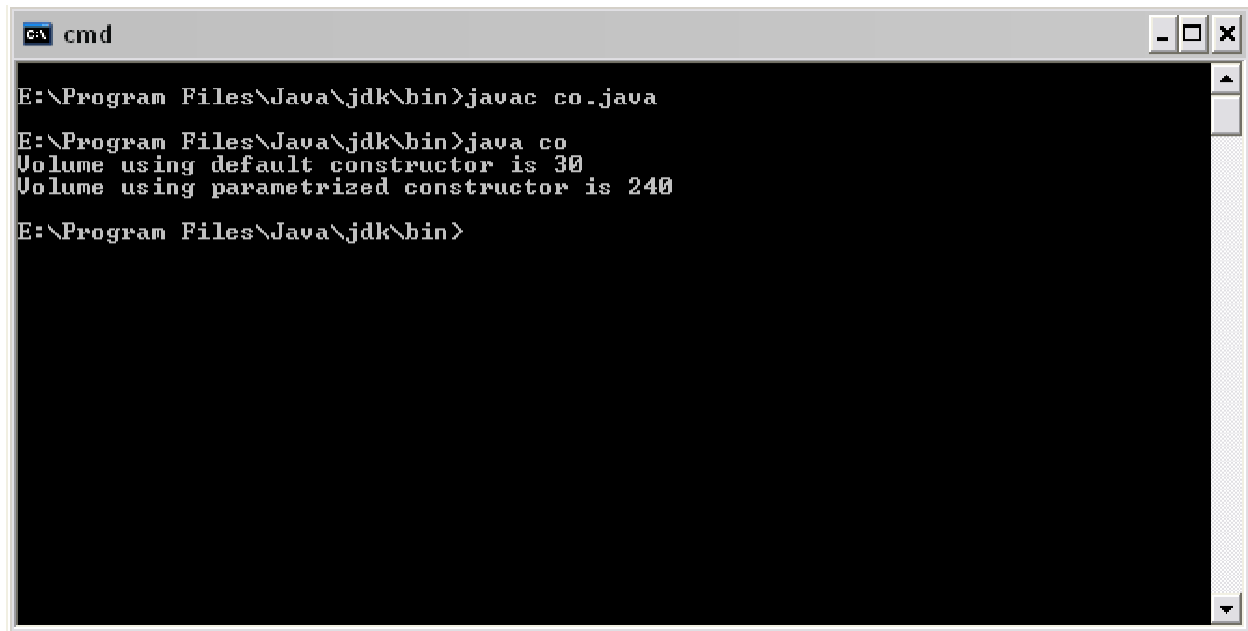
PROGRAM NO - 4

➢ Write a program in JAVA using constructor overloading to calculate volume.

```java
class const1
{
intl,b,h;
const1()
{
l=2;
b=3;
h=5;
}
const1(int l1,int b1,int h1)
{
l=l1;
b=b1;
h=h1;
}
}
class co
{
public static void main(String args[])
    {
    const1 v=new const1();
    const1 v1=new const1(3,8,10);int
    volume=v.l*v.b*v.h;
    int volume1=v1.l*v1.b*v1.h;
    System.out.println("Volume using default constructor is "+volume);
    System.out.println("Volume using parametrized constructor is

"+volume1);
```

```
        }
    }
```

OUTPUT
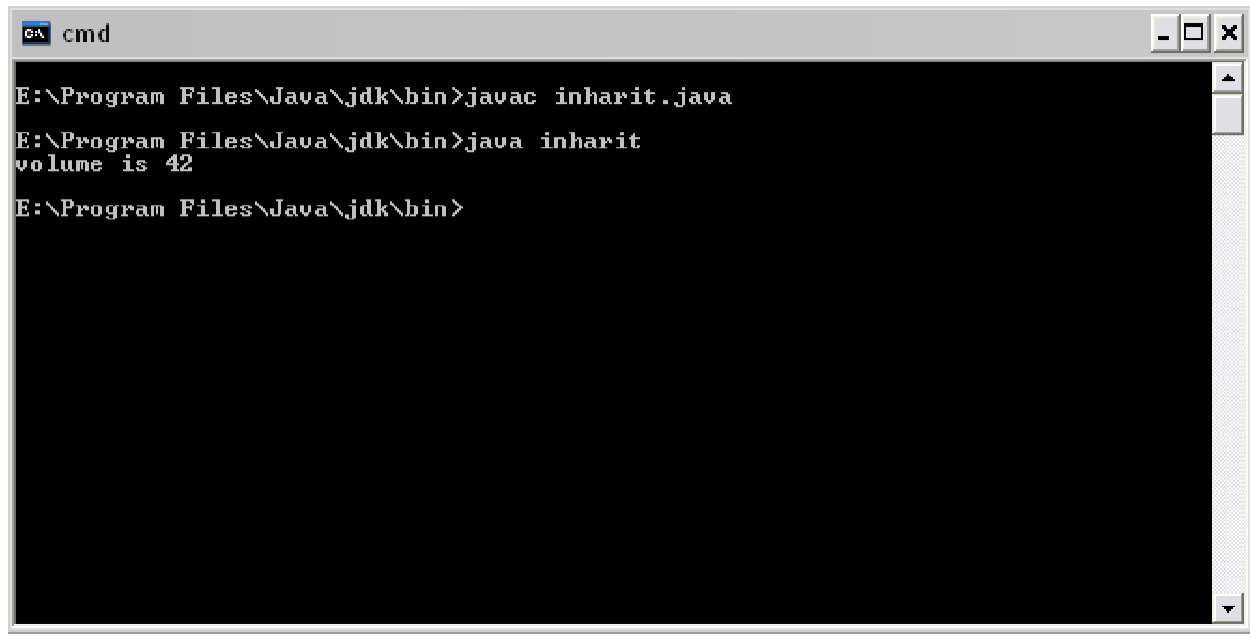
```
cmd                                                    _ □ ×

E:\Program Files\Java\jdk\bin>javac co.java

E:\Program Files\Java\jdk\bin>java co
Volume using default constructor is 30
Volume using parametrized constructor is 240

E:\Program Files\Java\jdk\bin>
```

PROGRAM NO – 5

➤ Write a program in JAVA to calculate volume using single inheritance.

```java
classabc

{intl,b,h;

abc()

{

l=2;

b=3;

h=7;

}}

class xyz extends abc

{

}

classinharit

{

public static void main(String args[])

{

xyz v=new xyz();

int volume=v.b*v.l*v.h;

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

}

}
```

OUTPUT

```
cmd                                                                    _ □ ✕

E:\Program Files\Java\jdk\bin>javac inharit.java

E:\Program Files\Java\jdk\bin>java inharit
volume is 42

E:\Program Files\Java\jdk\bin>
```
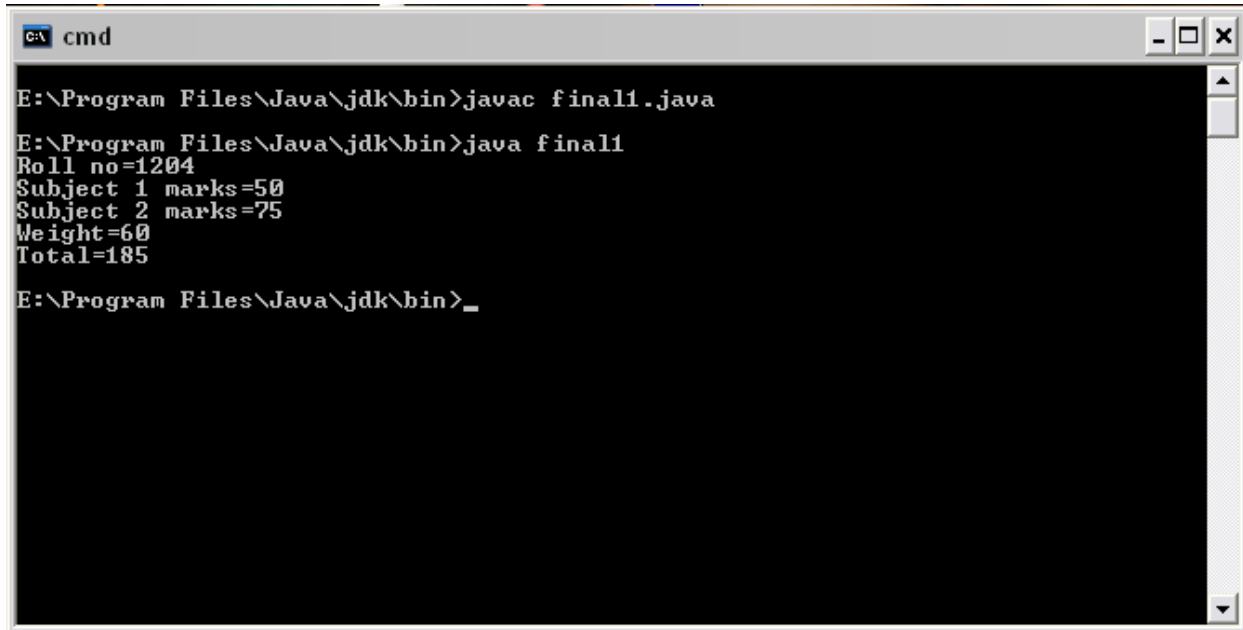
PROGRAM NO - 6

> Write a program in JAVA to implement multiple inheritance using Interface

```
class student{
introllno;
voidgetno(int r){
rollno=r;}
voidputno(){
System.out.println("Roll no="+ rollno);
}}
class marks extends student
{
int sub1,sub2;
voidgetmarks(int s1,int s2)
{
sub1=s1;
sub2=s2;
}
voidputmarks(){
System.out.println("Subject 1 marks="+sub1);
System.out.println("Subject 2 marks="+sub2);
}}
interface weight{
intwt=60;
voidputwt();
```

```java
}
class result extends marks implements weight
{
int total;
public void putwt()
{
System.out.println("Weight="+wt);}void
display(){ total=sub1+sub2+wt;
putno();
putmarks();
putwt();
System.out.println("Total="+total);}
}
class final1
{
public static void main(String args[]){result
r1=new result(); r1.getno(1204);
r1.getmarks(50,75);
r1.display();
}}
```

OUTPUT

```
cmd                                                    _ □ ✖

E:\Program Files\Java\jdk\bin>javac final1.java

E:\Program Files\Java\jdk\bin>java final1
Roll no=1204
Subject 1 marks=50
Subject 2 marks=75
Weight=60
Total=185

E:\Program Files\Java\jdk\bin>_
```

PROGRAM NO - 7

➢ Write a program to create and import a package to calculate marks and print the grade of student.

```
package pack; public

class student

{

int sub1,sub2;

public void getmarks(intx,int y)

{

sub1=x;

sub2=y;

}

public void putmarks()

{

int tot=sub1+sub2;

System.out.println("total="+tot);

}

}
```
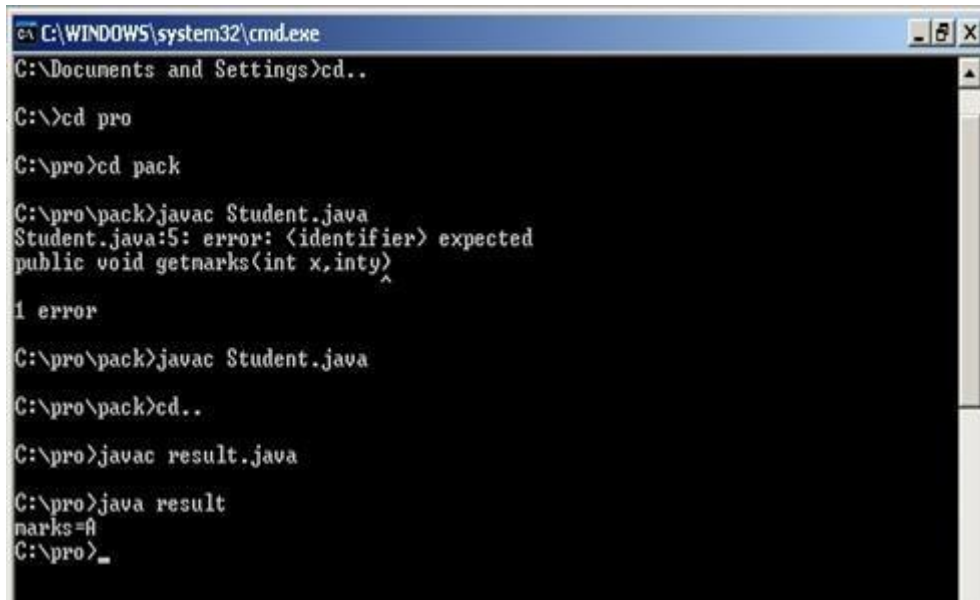
```
import pack.*;

class grade extends student

{

char gr;
```

```
grade(char p)

{

gr=p;

}

voidgetresult()

{

putmarks();

System.out.println("grade="+gr);

}

}

class result

{

public static void main(String args[])

{

grade r=new grade('A');

r.getmarks(60,80);

r.getresult();

}

}
```
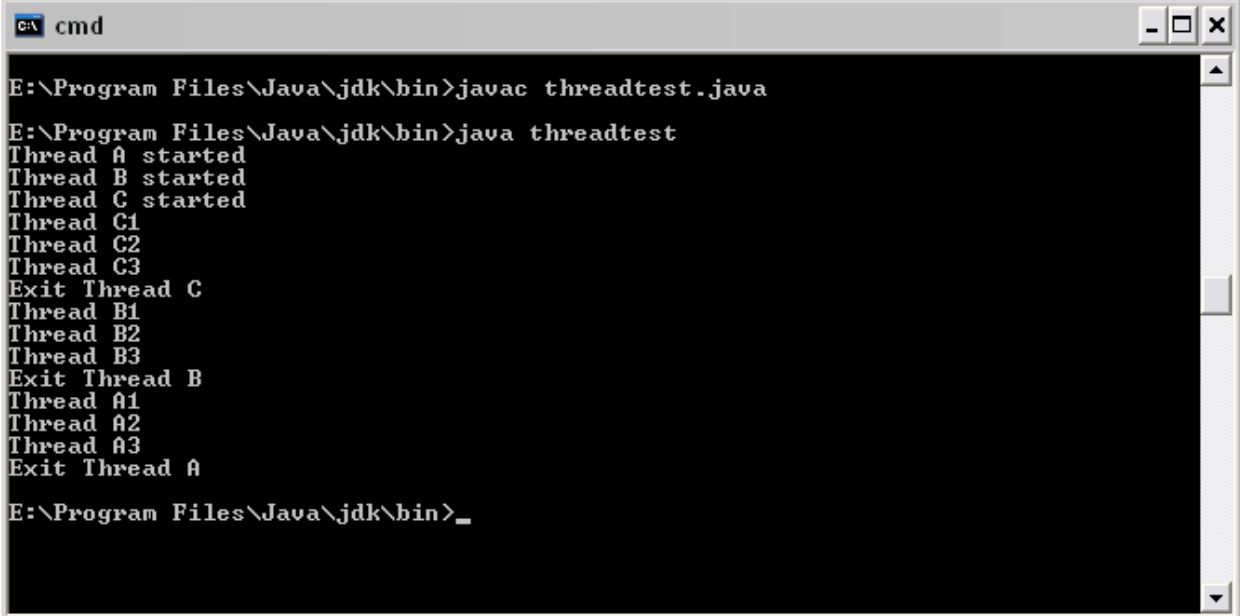
PROGRAM NO - 8

➢ Write a program to set the priority of a thread in Multithreading.

```java
class A extends Thread
{
public void run()
{
System.out.println("Thread A started");
for(inti=1;i<=3;i++)
{
System.out.println("Thread A"+i);
}
System.out.println("Exit Thread A");
}
}
class B extends Thread
{
public void run()
{
System.out.println("Thread B started");for(int
j=1;j<=3;j++)
{
System.out.println("Thread B"+j);
}
System.out.println("Exit Thread B");
}
}
class C extends Thread
{
public void run()
{
System.out.println("Thread C started");for(int
k=1;k<=3;k++)
{
System.out.println("Thread C"+k);
}
System.out.println("Exit Thread C");
}
```

```java
}
classthreadtest
{
public static void main(String args[])
{
A  tha=new  A();B
thb=new   B();   C
thc=new C();
thc.setPriority(Thread.MAX_PRIORITY);
thb.setPriority(tha.getPriority()+1);
tha.setPriority(Thread.MIN_PRIORITY);
tha.start();
thb.start();
thc.start();
}
}
```

OUTPUT

```
cmd                                                                    _ □ ×

E:\Program Files\Java\jdk\bin>javac threadtest.java

E:\Program Files\Java\jdk\bin>java threadtest
Thread A started
Thread B started
Thread C started
Thread C1
Thread C2
Thread C3
Exit Thread C
Thread B1
Thread B2
Thread B3
Exit Thread B
Thread A1
Thread A2
Thread A3
Exit Thread A

E:\Program Files\Java\jdk\bin>_
```
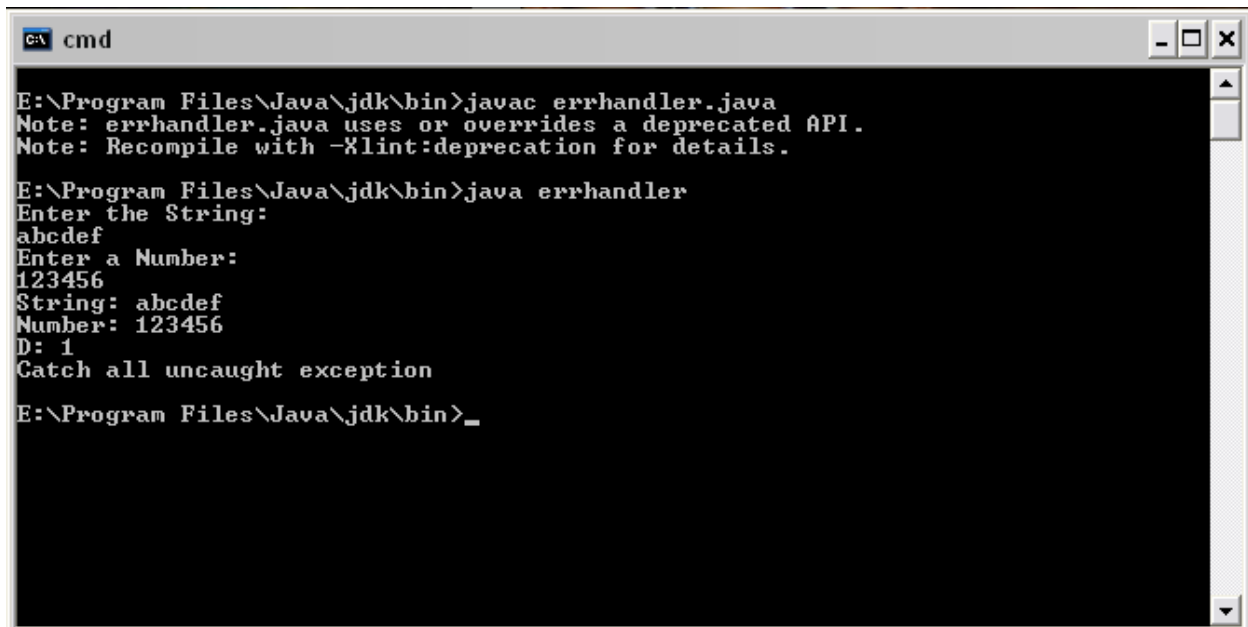
PROGRAM NO - 9

> ➤ Write a program for handling uncaught exception using finally.

```java
import java.io.*;

classerrhandler

{

public static void main(String args[])

{

try

{

DataInputStreambr=new DataInputStream(System.in);

System.out.println("Enter the String: ");

String s=br.readLine();

System.out.println("Enter a Number: ");

inti=Integer.parseInt(br.readLine());

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

System.out.println("Number: "+i);

int   a=10;

int   b=15;

int c=5;

int d=a/(b-c); System.out.println("D:

"+d);

}

catch(ArithmeticException e)
```

```java
{

System.out.println("Arithmetic Exception Catch");

}

catch(IOException e)

{

System.out.println("IOException Catch");

}

finally

{

System.out.println("Catch all uncaught exception");

}

}

}
```

OUTPUT

```
cmd                                                              _ □ ✕

E:\Program Files\Java\jdk\bin>javac errhandler.java
Note: errhandler.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

E:\Program Files\Java\jdk\bin>java errhandler
Enter the String:
abcdef
Enter a Number:
123456
String: abcdef
Number: 123456
D: 1
Catch all uncaught exception

E:\Program Files\Java\jdk\bin>_
```

<u>PROGRAM NO - 10</u>

➢ Write a program to show a face on an Applet.

```java
/* <applet

code=face.class

width=300

height=300

>

</applet>*/

importjava.awt.*;

importjava.applet.*;

public class face extends Applet

{

public void paint(Graphics g)

{

g.drawOval(40,40,120,150);

g.setColor(Color.blue);

g.drawOval(57,75,30,20);

g.drawOval(110,75,30,20);

g.fillOval(68,81,10,10);
```

```
g.fillOval(121,81,10,10);

g.drawOval(85,100,30,30);

g.setColor(Color.red);

g.fillArc(60,125,80,40,180,180);

g.drawOval(25,92,15,30);

g.drawOval(160,92,15,30);

}
```