ANJUMAN
COLLEGE OF ENGINEERING & TECHNOLOGY
(MANAGED BY : ANJUMAN HAMI-E-ISLAM, NAGPUR)

Object Oriented Programming with Java Lab Manual

Computer Science & Engineering Department

Roll No:____
Name:_____
Sem:_____Section_____

CERTIFICATE

Certified that this file is submitted by

Shri/Ku._____

Roll No._____a student of _____ year of the course _____

_____ as a part of PRACTICAL/ORAL as prescribed by the Rashtrasant Tukadoji Maharaj Nagpur University for the subject_____ in the laboratory of _____during the academic year _____ and that I have instructed him/her for the said work, from time to time and I found him/her to be satisfactory progressive.

And that I have accessed the said work and I am satisfied that the same is up to that standard envisaged for the course.

Date:- Signature & Name of Subject Teacher Signature & Name of HOD

Anjuman College of Engineering and Technology

Vision

- To be a centre of excellence for developing quality technocrats with moral and social ethics, to face the global challenges for the sustainable development of society.

Mission

- To create conducive academic culture for learning and identifying career goals.
- To provide quality technical education, research opportunities and imbibe entrepreneurship skills contributing to the socio-economic growth of the Nation.
- To inculcate values and skills, that will empower our students towards development through technology.

Vision and Mission of the Department

Vision:

- To achieve excellent standards of quality education in the field of computer science and engineering, aiming towards development of ethically strong technical experts contributing to the profession in the global society.

Mission:

- To create outcome based education environment for learning and identifying career goals.
- Provide latest tools in a learning ambience to enhance innovations, problem solving skills, leadership qualities team spirit and ethical responsibilities.
- Inculcating awareness through innovative activities in the emerging areas of technology.

Program Educational Objectives (PEOs)

- The graduates will have a strong foundation in mathematical, scientific and engineering fundamentals necessary to formulate, solve and analyze engineering problem in their career.
- Graduates will be able to create and design computer support systems and impart knowledge and skills to analyze, design, test and implement various software applications.
- Graduates will work productively as computer science engineers towards betterment of society exhibiting ethical qualities.

Program Specific Outcomes (PSOs)

- Foundation of mathematical concepts: To use mathematical methodologies and techniques for computing and solving problem using suitable mathematical analysis, data structures, database and algorithms as per the requirement.
- Foundation of Computer System: The capability and ability to interpret and understand the fundamental concepts and methodology of computer systems and programming. Students can understand the functionality of hardware and software aspects of computer systems, networks and security.
- Foundations of Software development: The ability to grasp the software development lifecycle and methodologies of software system and project development.

PROGRAM: CSE	DEGREE: B.E
COURSE: OBJECT ORIENTED PROGRAMMING WITH JAVA LAB MANUAL	SEMESTER: III CREDITS: 2
COURSE CODE: BECSE302T	COURSE TYPE: REGULAR
COURSE AREA/DOMAIN: : OBJECT ORIENTED PROGRAMMING	CONTACT HOURS: 2 hours/Week.
CORRESPONDING LAB COURSE CODE : BECSE302P	LAB COURSE NAME : OBJECT ORIENTED PROGRAMMING WITH JAVA LAB

COURSE PRE-REQUISITES:

C.CODE	COURSE NAME	DESCRIPTION	SEM
BESI-8	C, C++	Knowledge of structured programming language & application development.	II

LAB COURSE OBJECTIVES:

- Gain knowledge about basic Java language syntax and semantics to write Java programs and use concepts such as variables, conditional and iterative execution methods.
- Be able to use the Java SDK environment to create, debug, and run simple Java programs.
- To analyze the object-oriented paradigm using java programming language.
- To implement small/medium scale java programs to resolve small business problems.

COURSE OUTCOMES:

After completion of this course the students will be able -

SNO	DESCRIPTION	BLOOM'S TAXONOMY LEVEL
CO.1	**Identify** classes, objects and relationship among them for a specific problem.	LEVEL I
CO.2	**Apply** the concepts of garbage collection, polymorphism, inheritance etc	LEVEL III
CO.3	**Apply** numeric (algebraic) and string based computation.	LEVEL III
CO.4	**Develop** modularity as well as basic error handling techniques.	LEVEL VI
CO.5	**Write** small multithreaded programs using Java language.	LEVEL VI
CO.6	**Apply** appropriate problem solving strategies for the implementation of small- medium scale java application.	LEVEL III

Lab Instructions:

- Make entry in the Log Book as soon as you enter the Laboratory.
- All the students should sit according to their Roll Numbers.
- All the students are supposed to enter the terminal number in the Log Book.
- Do not change the terminal on which you are working.
- Strictly observe the instructions given by the Faculty / Lab. Instructor.
- Take permission before entering in the lab and keep your belongings in the racks.
- NO FOOD, DRINK, IN ANY FORM is allowed in the lab.
- TURN OFF CELL PHONES! If you need to use it, please keep it in bags.
- Avoid all horseplay in the laboratory. Do not misbehave in the computer laboratory. Work quietly.
- Save often and keep your files organized.
- Don't change settings and surf safely.
- Do not reboot, turn off, or move any workstation or PC.
- Do not load any software on any lab computer (without prior permission of Faculty and Technical Support Personnel). Only Lab Operators and Technical Support Personnel are authorized to carry out these tasks.
- Do not reconfigure the cabling/equipment without prior permission.
- Do not play games on systems.
- Turn off the machine once you are done using it.
- Violation of the above rules and etiquette guidelines will result in disciplinary action.

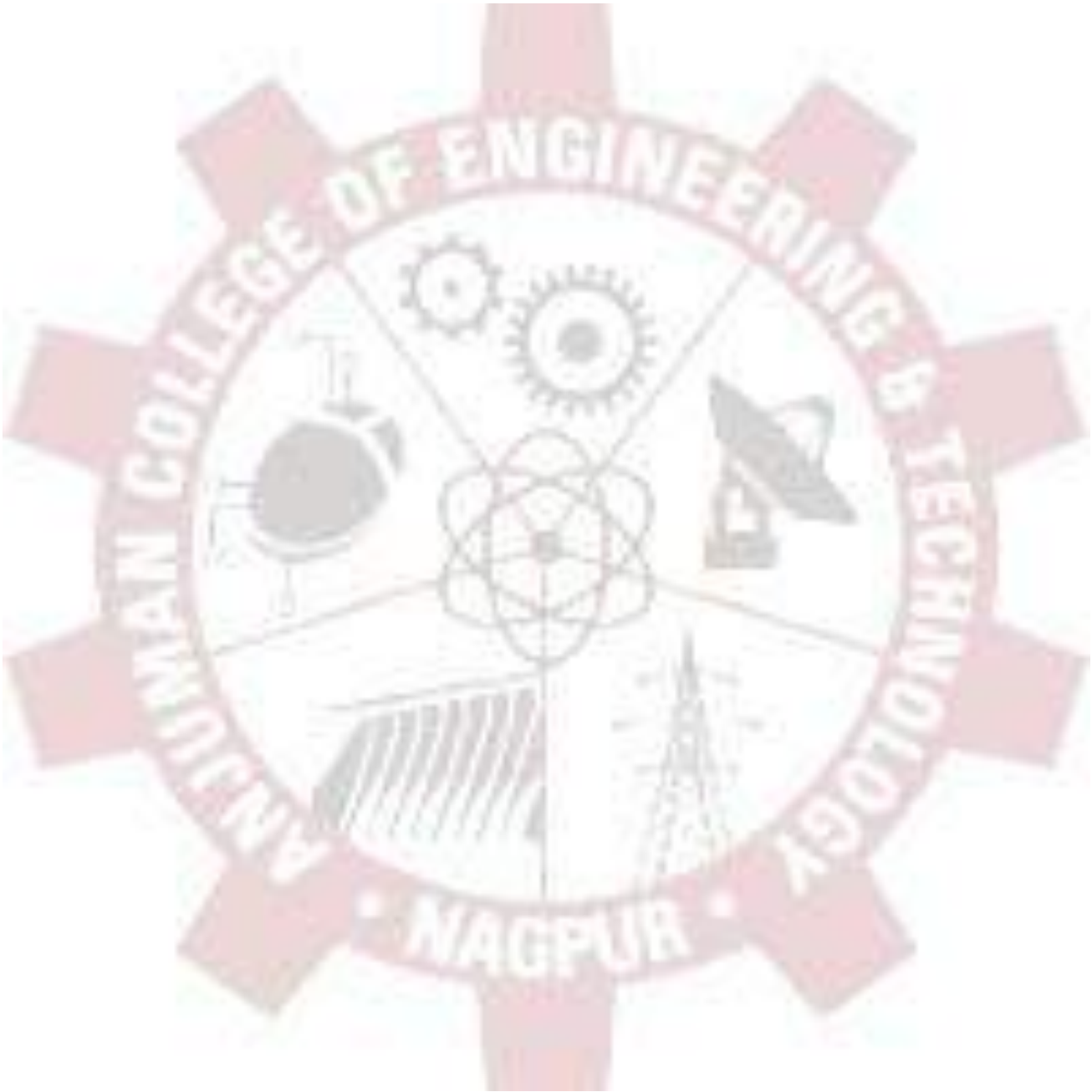
Continuous Assessment Practical

Exp No	NAME OF EXPERIMENT	Date	Sign	Remark
1	a. Write a program to print the area of triangle. Save it with name Area.java in your folder .class b. Write a java Program to check the number is Prime or not. c. Write a java Program to generate a Ladder of number.			
2	a. Write a program to create a class Student with data 'name, city and age' along with method printData to display the data. Create the two objects s1 ,s2 to declare and access the values. b. Write a program to create a class Student2 along with two method getData(),printData() to get the value through argument and display the data in printData. Create the two objects s1 , s2 to declare and access the values from class STtest. c. WAP using parameterized constructor with two parameters id and name. While creating the objects obj1 and obj2 passed two arguments so that this constructor gets invoked after creation of obj1 and obj2.			
3	a. Write a Java Program to finds the average of numbers in an array. b. Write a Java Program to finds addition of two matrices.			
4	a. Java Program to convert all primitives into its corresponding .wr apper objects and vice-versa . b. Creating the custom wrapper class.			
5	a. Java program to illustrate the concept of Multilevel inheritance b. A Simple Java program to demonstrate method overriding in java. c. Write a java program in which you will declare an abstract class Vehicle inherits this class from two classes car and truck using the method engine in both display "car has good engine" and "truck has bad engine".			
6	a. Write a program in java if number is less than 10 and greater than 50 it generate the exception out of range. Else it displays the square of number. b. Write a program in java to enter the number through command line argument if first and second number is not entered it will generate the exception. Also divide the first number with second number and generate the arithmetic exception. c. Write a program in java to enter the number through command line argument if first and second number .using the method divides the first number with second and generate the exception.			

7	a. Write a java program in which thread sleep for 5 sec and change the name of thread. b. Write a java program in which thread sleep for 6 sec in the loop in reverse order from 5 to 1 and change the name of thread.			
8	a. Write a java program for multithread in which user thread and thread started from main method invoked at a time each thread sleep for 1 sec. b. Write a java program for to solve producer consumer problem in which a producer produce a value and consumer consume the value before producer generate the next value.			
9	Java programe on array list.			
10	Programe to demonstrate HashMap			

CONTENTS

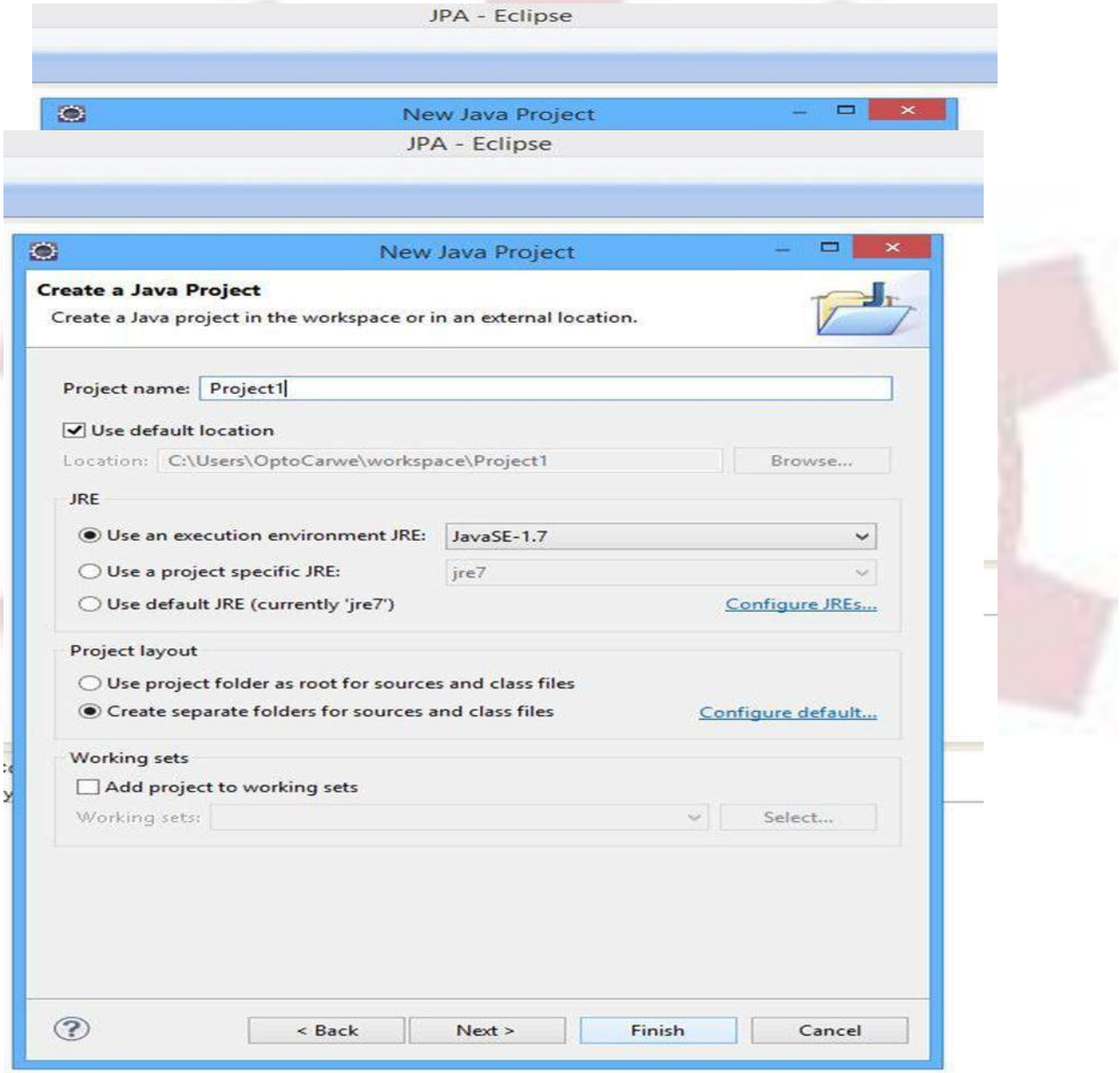
	producer produce a value and consumer consume the value before producer generate the next value.	
9	Java programe on array list.	
10	Programe to demonstrate HashMap	



Steps To Execute Simple Java Program Using Eclipse

Step1: Begin by creating a new Java project.

There are few different ways of accomplishing this. Click the arrow next to the left-most icon on the toolbar and select “Project” from the drop-down menu. Alternately Start a new Java Project by choosing “File” then “New” followed by “Java Project”. Also use the shortcut Alt+Shift+N.

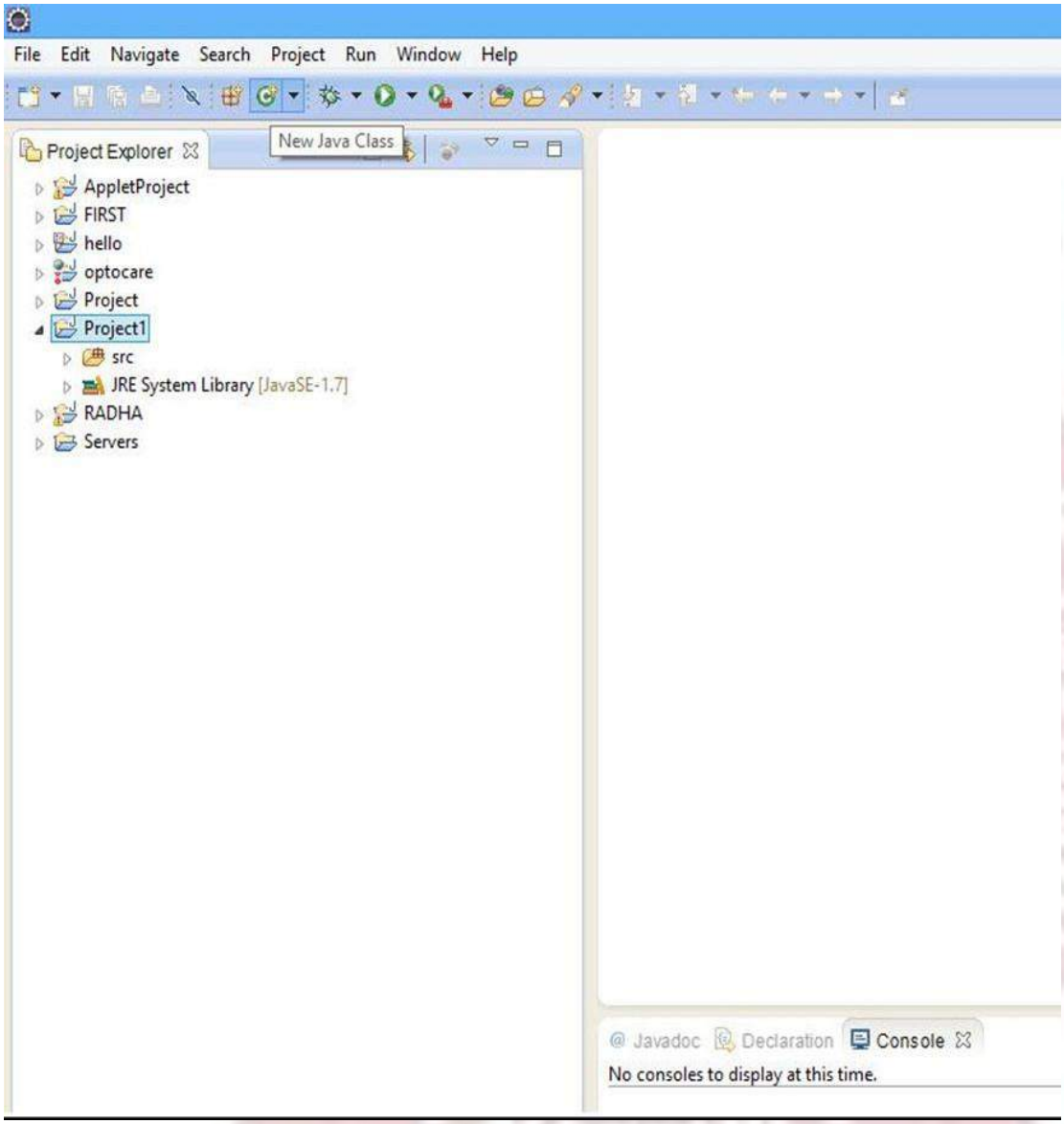


Step2: Enter a Project Name :

You will see a window titled “Create a Java Project”. The buttons “Next” and “Finish” at the bottom of the window will be grayed out until a project name is entered in the first field. To processed, give project name and enter it into this field then click “Finish”. New project will appear on the left-hand side of the screen under “Package Explorer” among existing projects. Projects are listed in alphabetical order.

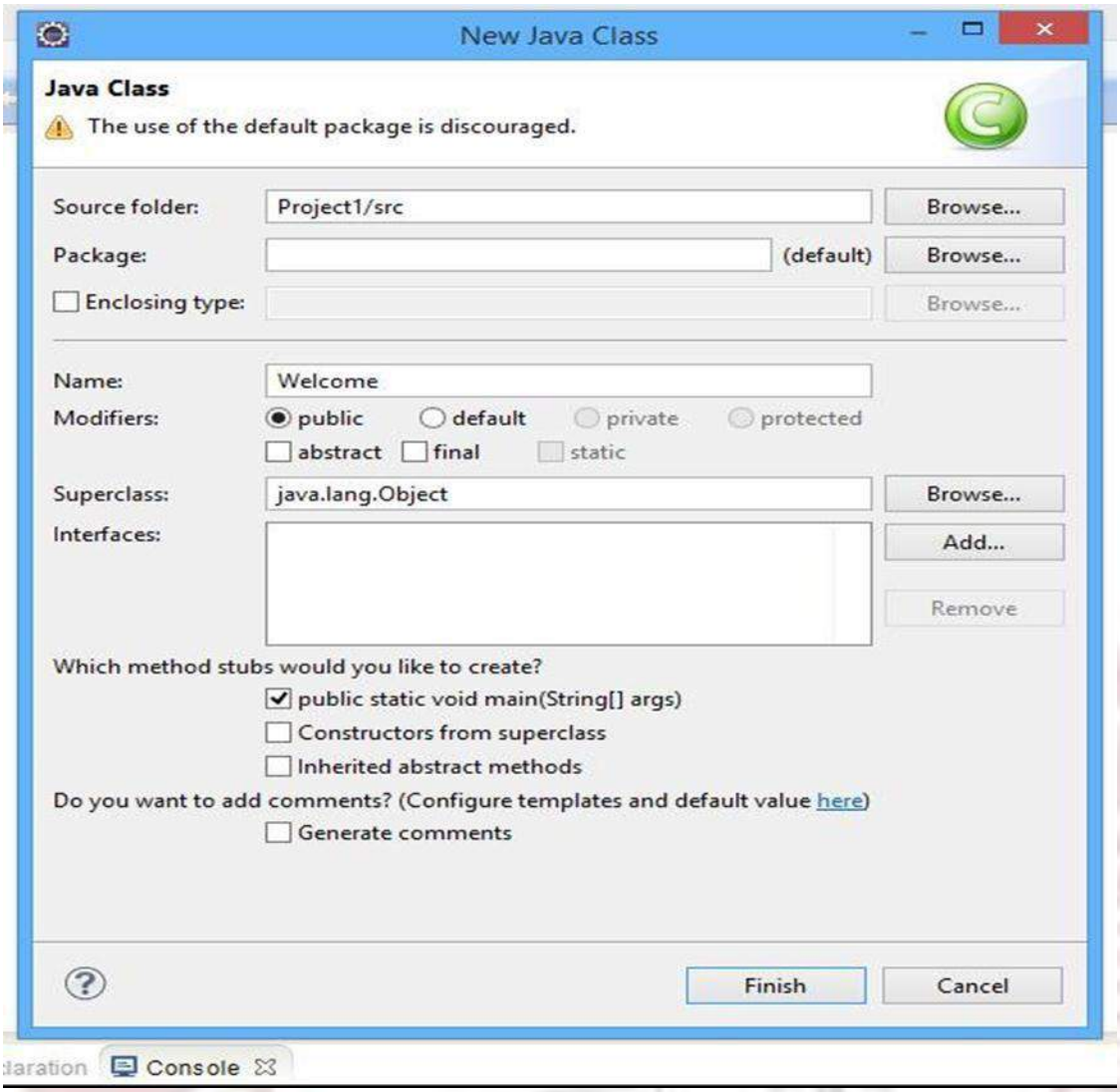
Step3: Start a new java class.

Before begin writing code, need to create a new Java class. A class is a blueprint for an object. It defines the data stored in the object as well as its actions. Create a class by clicking the "New Java Class" icon, which looks like a green circle with the letter "C" in the center of it.



4: Enter the name of your class.

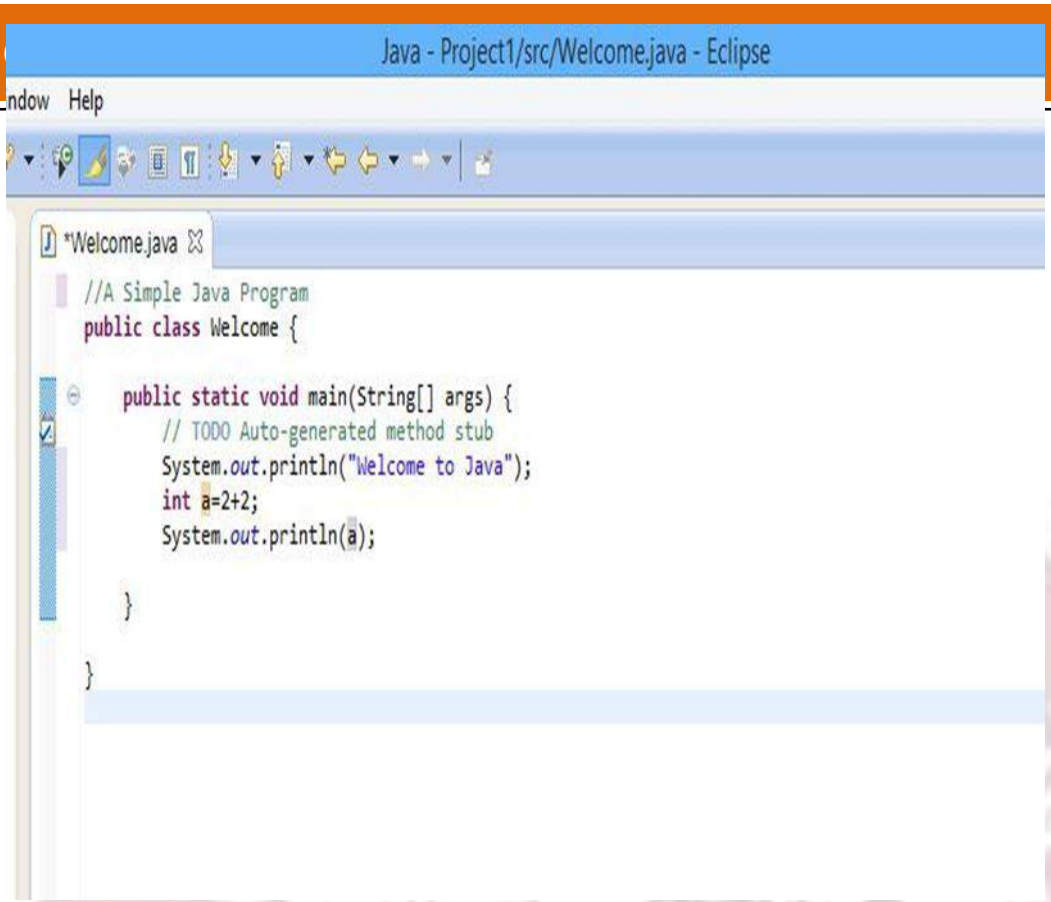
You will see a window titled "Java Class." To proceed, enter the name of class into the field "Name". Since the class will be main class of the simple project, check theselection box labeled "public static void main(String[] args)" to include the method stub. Afterwards, click "Finish".



Step 5: Enter Java Code.

Here new class Welcome.java is created. It appears with the method stub "public static void main(String[] args)" along with some automatically generated comments. A

method will contain a sequence of instructions to be executed by the program. A comment is a statement that is ignored by the compiler. Comments are used by programmers to document their code. Edit this file and insert the code for Java Program.



Step 6: Watch out for errors in code.

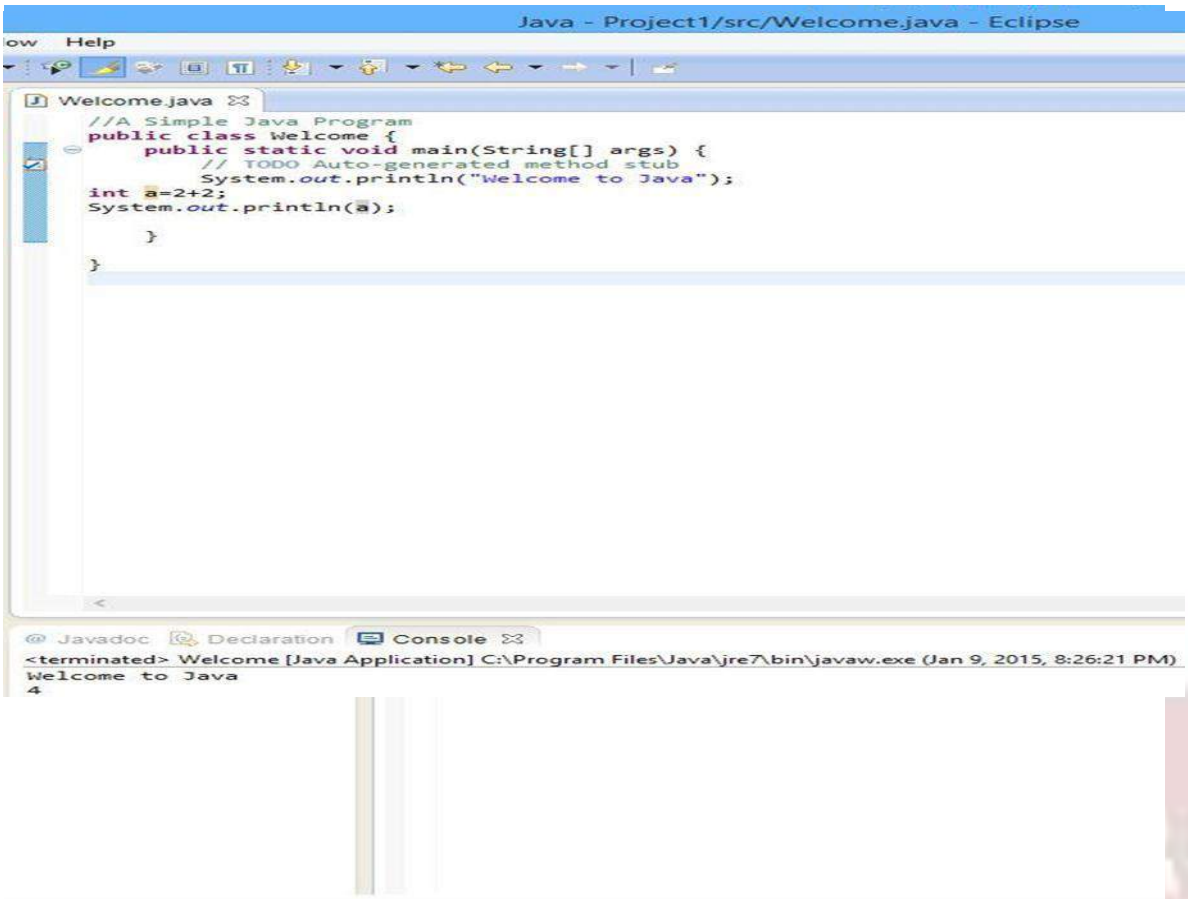
Any errors will be underlined in red, and icon with an ”X” will show up on the left. Fix errors. By mousing over an error icon. Can see a suggestion box that lists the ways can fix the error.

Step 7: Ensure that entire program is free of errors.

There are three types of errors must beware of: syntax errors, run-time errors and logic errors. The compiler will alert syntax errors. Examples of syntax errors are misspelled variable names or missing semi-colons. Until remove all syntax errors from code program will not compile. The compiler will not catch run-time errors or logic errors.

Step 8: Compile Java Program.

Now the program is free for errors, click the triangular icon to run program. Another way to run program is to select “Run” from the main menu and then select “Run” again from the drop-down menu. The shortcut is Ctrl+F11.



Step 9: Verify the output is what you expected.

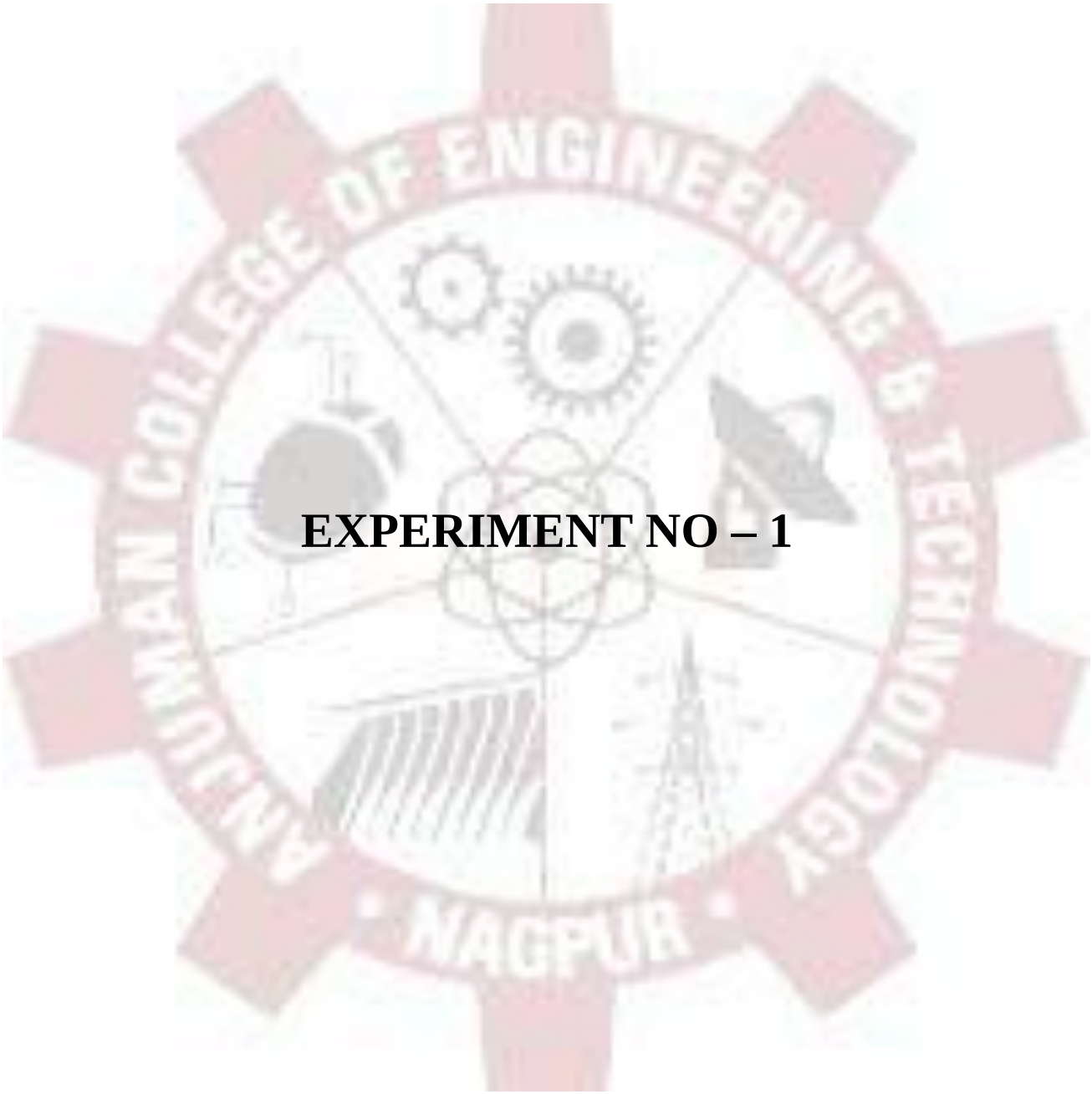
When program runs, the output will be displayed on console at the bottom of the screen.

Step 10: Fix any run-time or logic errors.

If the output is different from what you excepted, then there might have been an error even though the program compiled. For example, if the output was zero instead of four, then there was a mistake in the program's calculation.

Example: Write a Program to print the text "Welcome to World of Java". Save it with name Welcome.java in your folder.

```
Class Welcome
{
public static void main (String args[])
{
System.out.println ("welcome to world of Java");
}
}
```



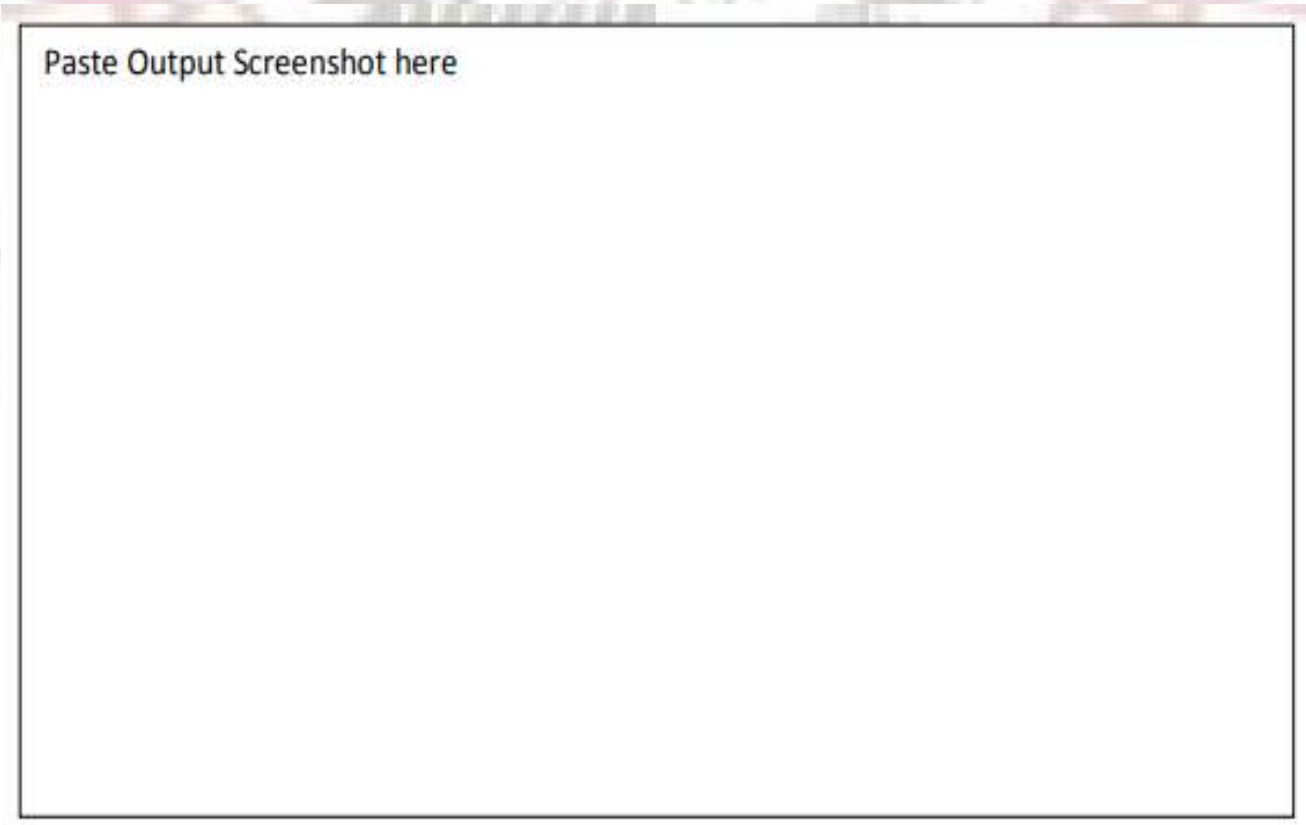
EXPERIMENT NO – 1

Aim: 1.a Write a program to print the area of triangle. Save it with name Area.java in your folder .class

Program:

```
Area
{
public static void main(String args[])
{
int height =10, base=6;
float area=0.5F*base* height;
System.out.println("area of triangle = "+area);
}
}
```

Output:

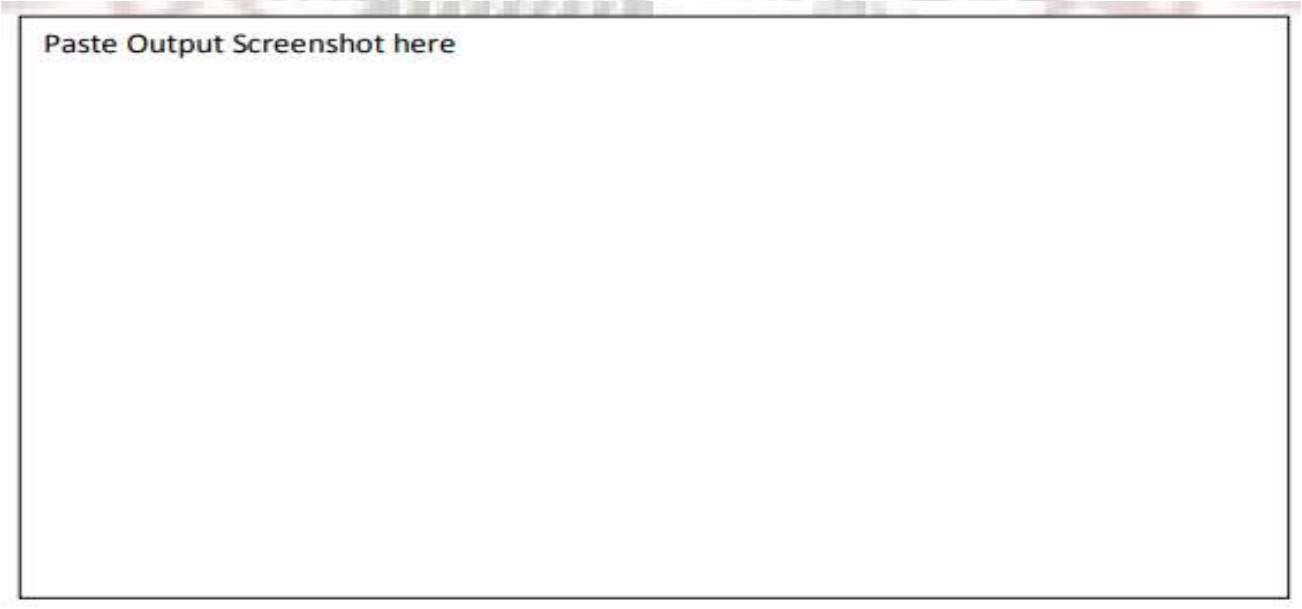


1b: Write a java program to check the number is Prime or not. Import java.util.Scanner;

Program:

```
Import java.util.Scanner;
class Prime
{
public static void main(String arr[])
{
int c;
Scanner in=new Scanner(System.in);
 System.out.println("Enter the number to be tested for prime ");
int n=in.nextInt();
for ( c = 2 ; c <= n - 1 ; c++ )
{
if ( n%c == 0 )
{
System.out.println(n+">>>>> not prime");
 break;
}
}
if ( c == n )
System.out.println(n+ ">>>>Number is prime.");

}
}
```

Output:



1c: Write a java Program to generate a Ladder of number.

Program:

```
Import java.util.Scanner;
class Ladder
{
public static void main(String arr[])
{
Scanner in=new Scanner(System.in);
System.out.println("Enter the number of rows");
int a=in.nextInt();
for(int i=1;i<=a;i++)
{
for(int j=1;j<=i;j++)
{
System.out.print(j);
for(int k=i-1;k>=1;k--)
System.out.print(k);
System.out.print("\n");
}
}
}
```

Output:

Paste Output Screenshot here

Viva Voce Question

1. Explain JDK, JRE and JVM?

_____

_____

_____

_____

2. Explain public static void main(String args[]) in Java?

_____

_____

_____

_____

3. Why Java is platform independent?

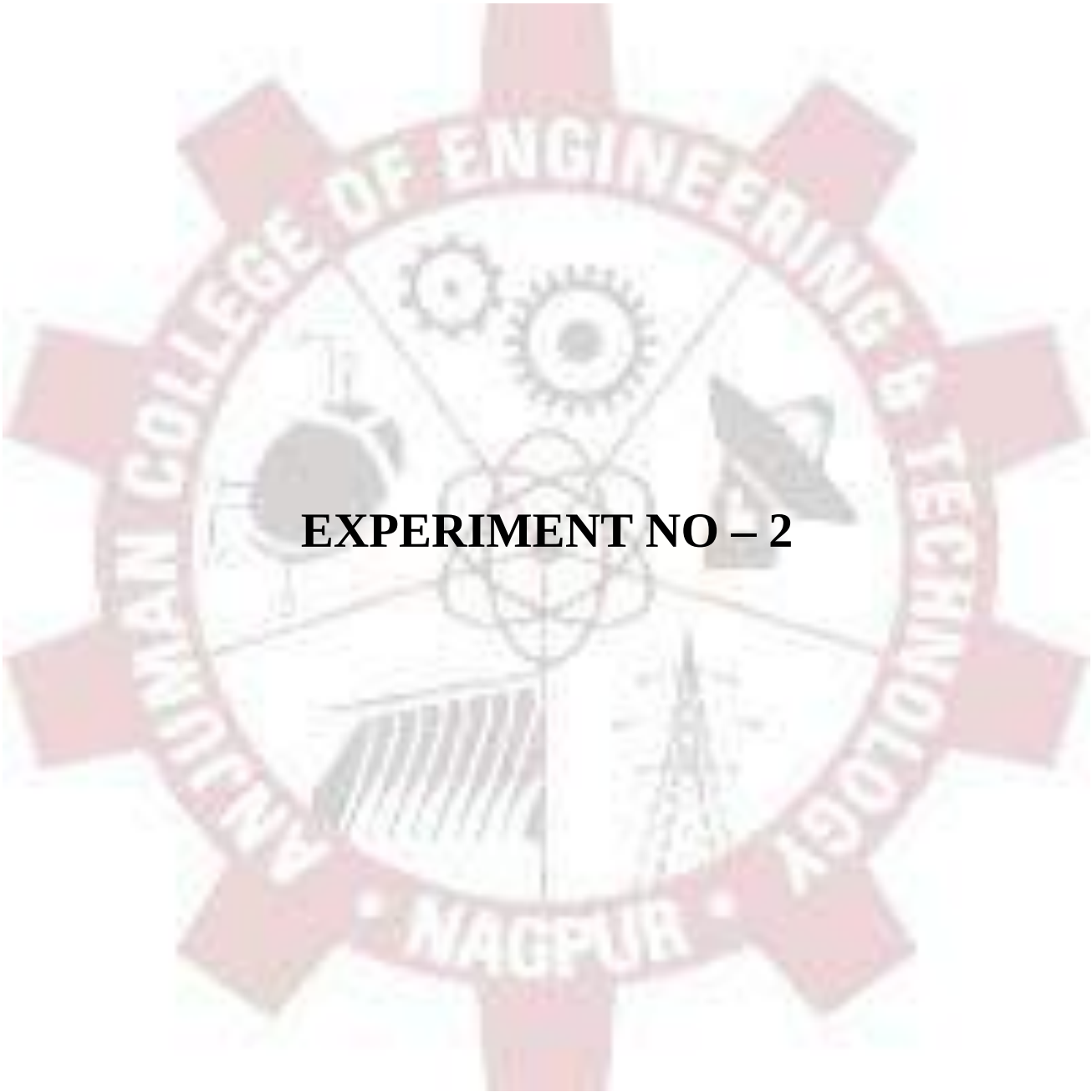
_____

_____

_____

_____

Signature of Subject Teacher



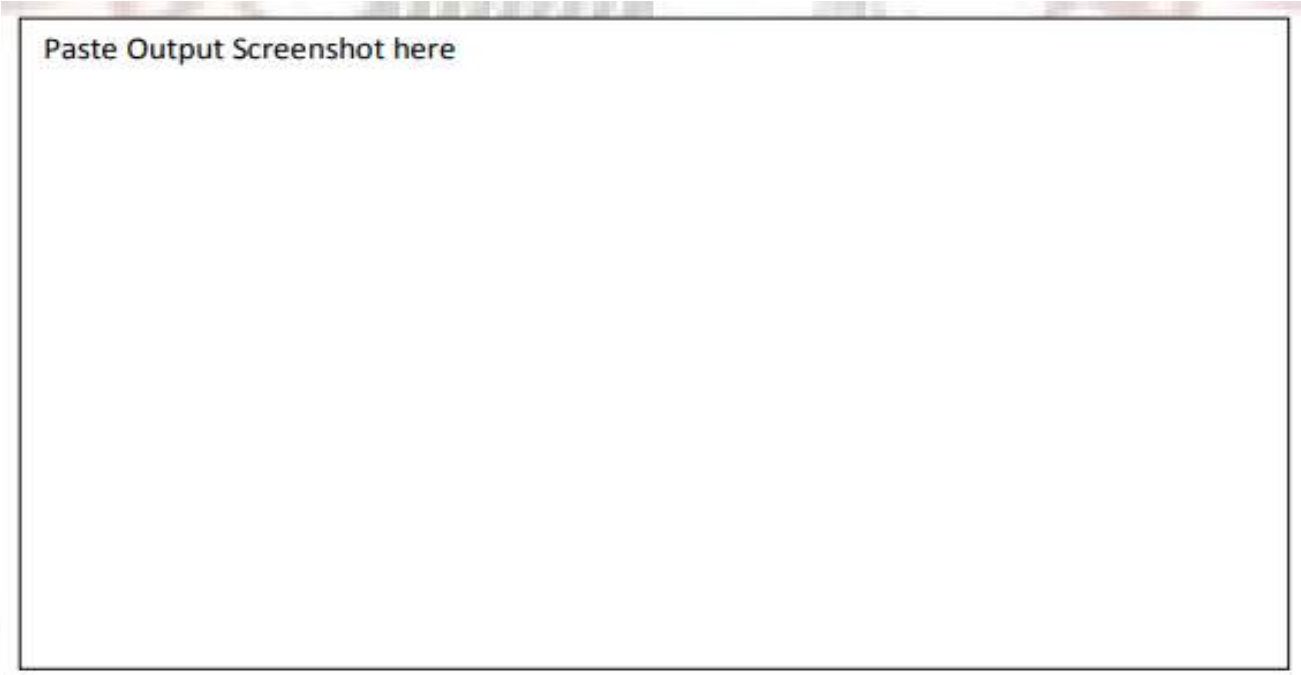
EXPERIMENT NO – 2

Aim: 2 a) Write a program to create a class Student with data 'name, city and age' along with method printData to display the data. Create the two objects s1 , s2 to declare and access the values.

Program:

```
class Student
{
String name, city;
int age;
static int m;
void printData()
{
System.out.println("Student name = "+name);
System.out.println("Student city = "+city);
System.out.println("Student age = "+age);
}
}
Class Stest
{
public static void main(String args[])
{
Student s1=new Student();
Student s2=new Student();
s1.name="Amit";
s1.city="Dehradun";
s1.age=22;
s2.name="Kapil";
s2.city="Delhi";
s2.age=23;
s2.printData();
 s1.printData();
 s1.m=20;
s2.m=22;
Student.m=27;
System.out.println("s1.m = "+s1.m);
System.out.println("s2.m = "+s2.m);
System.out.println("Student.m ="+Student.m);
}
}
```

Output:



2.b)Write a program to create a class Student2 along with two method getData(), printData() to get the value through argument and display the data in printData. Create the two objects s1 , s2 to declare and access the values from class STtest.

Program:

```
class Student2
{
private String name, city;
private int age;
public void getData(String x, Stringy, int t)
{
name=x;
city=y; age=t;
}
public void printData()
{
System.out.println("Student name ="+name);
 System.out.println("Student city ="+city);
System.out.println("Student age ="+age);
}
}
Class STtest
{
public static void main(String args[])
{
Student2 s1=new Student2();
Student2 s2=new Student2();
s2.getData("Kapil","Delhi",23);
s2.printData();
s1.getData("Amit","Dehradun",22);
 s1.printData();
}}
```

Output:

Paste Output Screenshot here

2.c) Write a program using parameterized constructor with two parameters id and name. While creating the objects obj1 and obj2 passed two arguments so that this constructor gets invoked after creation of obj1 and obj2.

Program:

```
class Employee
{
Int empId;
String empName;
//parameterized constructor with two parameters
Employee(int id, String name)
{
this.empId = id;
this.empName = name;
}
void info()
{
System.out.println("Id: "+empId+" Name: "+empName);
}
public static void main(String args[])
{
Employee obj1 = new Employee(10245,"Chaitanya");
 Employee obj2 = new Employee(92232,"Negan");
obj1.info();
obj2.info();
}
}
```

Paste Output Screenshot here

Viva Voce Question

1. What is a Constructor?

_____
_____
_____
_____
_____

2. Can we have a class with no Constructor in it? What will happen during object creation?

_____
_____
_____
_____
_____

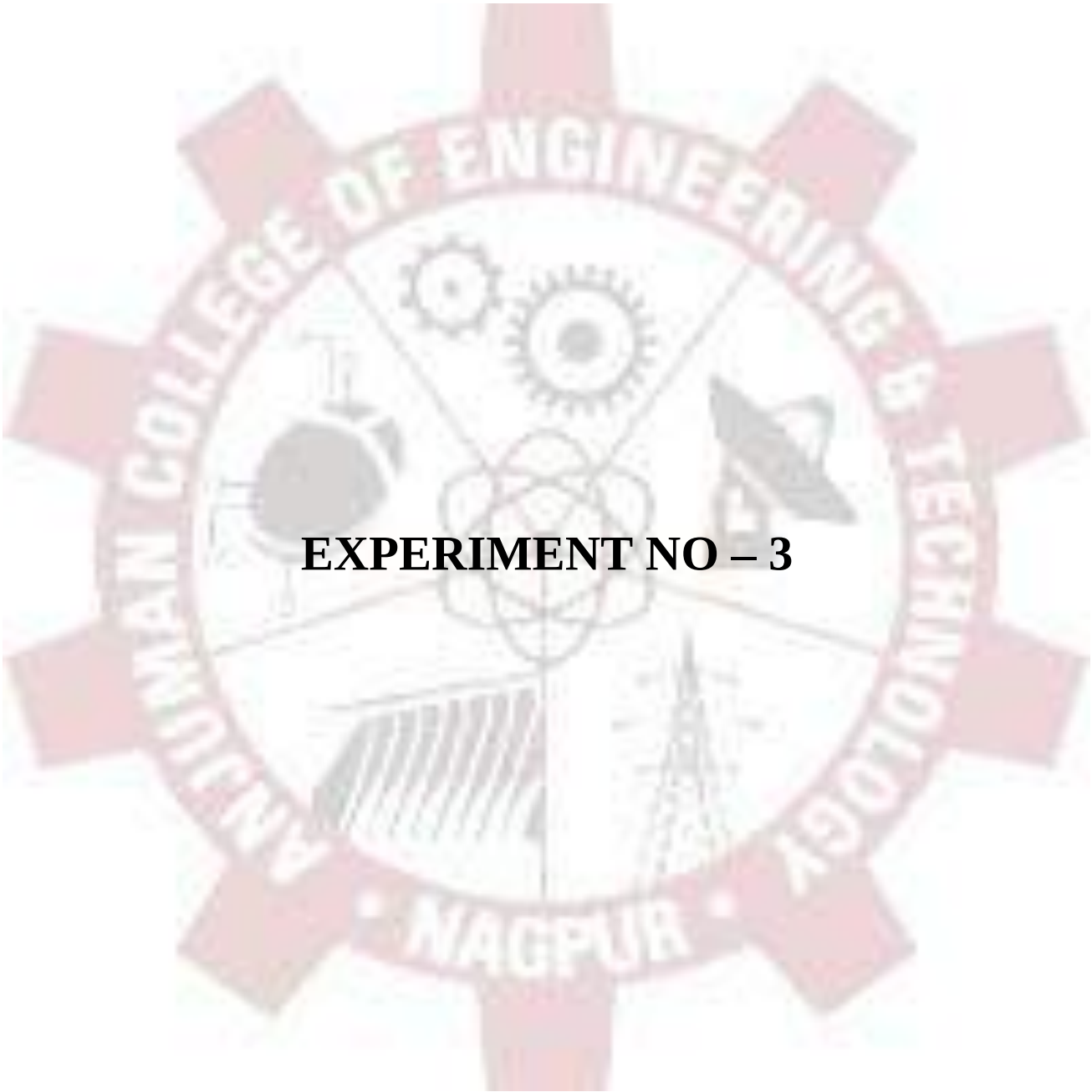
3. What is No-arg constructor??

_____
_____
_____
_____
_____

4. If we provide any arguments on the command line, then what will the value stored in the String array passed into the main() method, empty or NULL?

_____
_____
_____
_____
_____

Signature of Subject Teacher

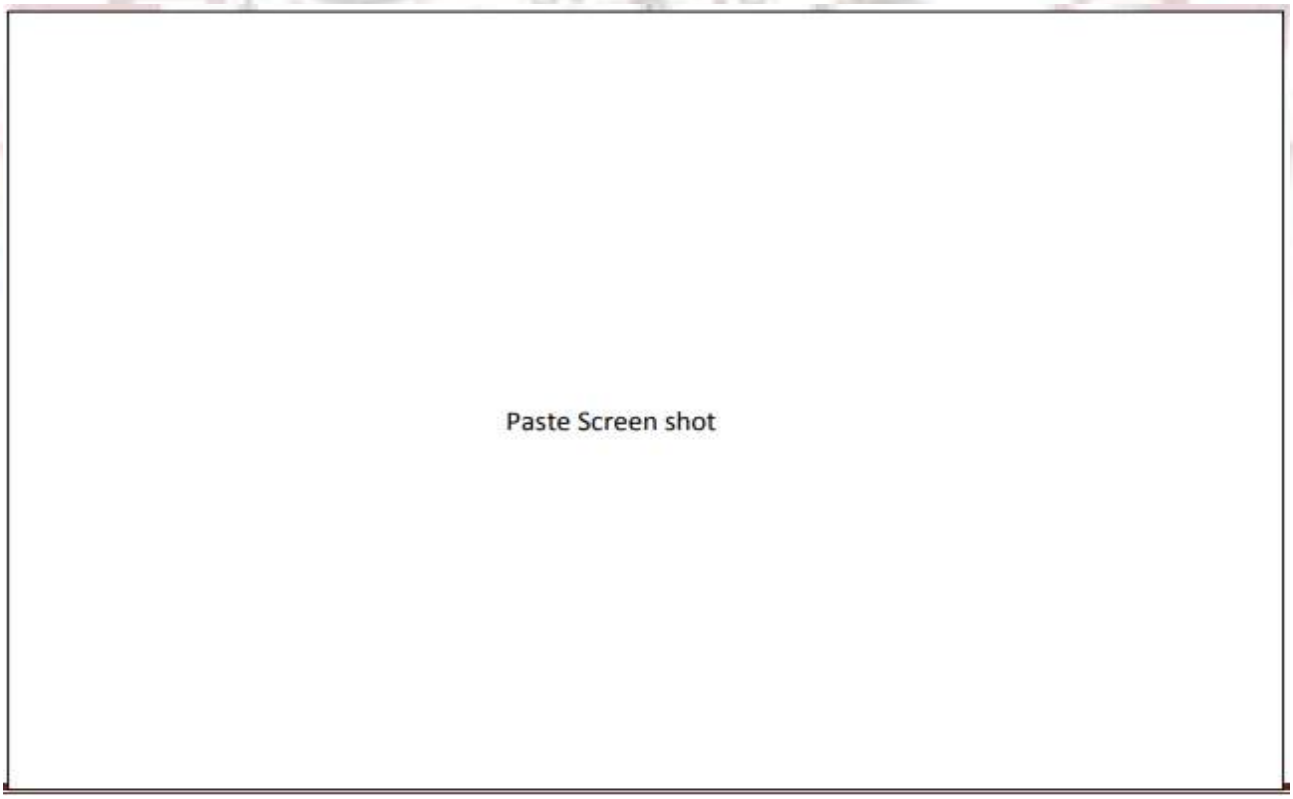


EXPERIMENT NO – 3

Aim:3 a) Write a Java Program to finds the average of numbers in an array.

Program:

```
Class Avg
{
public static void main(String args[])
{
int n=args.length;
float [] x=new float[n];
for(int i=0; i<n; i++)
{
x[i]=Float.parseFloat( args[i]);
}
float sum=0;
for(int i=0; i<n; i++)
sum=sum+x[i];
 float avg=sum/n;
System.out.println("Average of given numbers is "+avg);
}
}
```



3 b) Write a Java Program to finds addition of two matrices.

Program:

```
class Add
{
public static void main(String args[])
{
int [][] x={{1,2,3},{4,5,6},{7,8,9}};
int [][] y={{11,12,13},{14,15,16},{17,18,19}};
int [][] z=new int[3][3];
 for(int i=0; i<3; i++)
for(int j=0; j<3; j++)
{
z[i][j]=x[i][j]+y[i][j];
}
for(int i=0; i<3; i++)
{
for(int j=0; j<3; j++)
{
System.out.print(z[i][j]+" ");
}
System.out.print("\n");
}
}
}
```

Paste Screen shot

Viva Voce Question

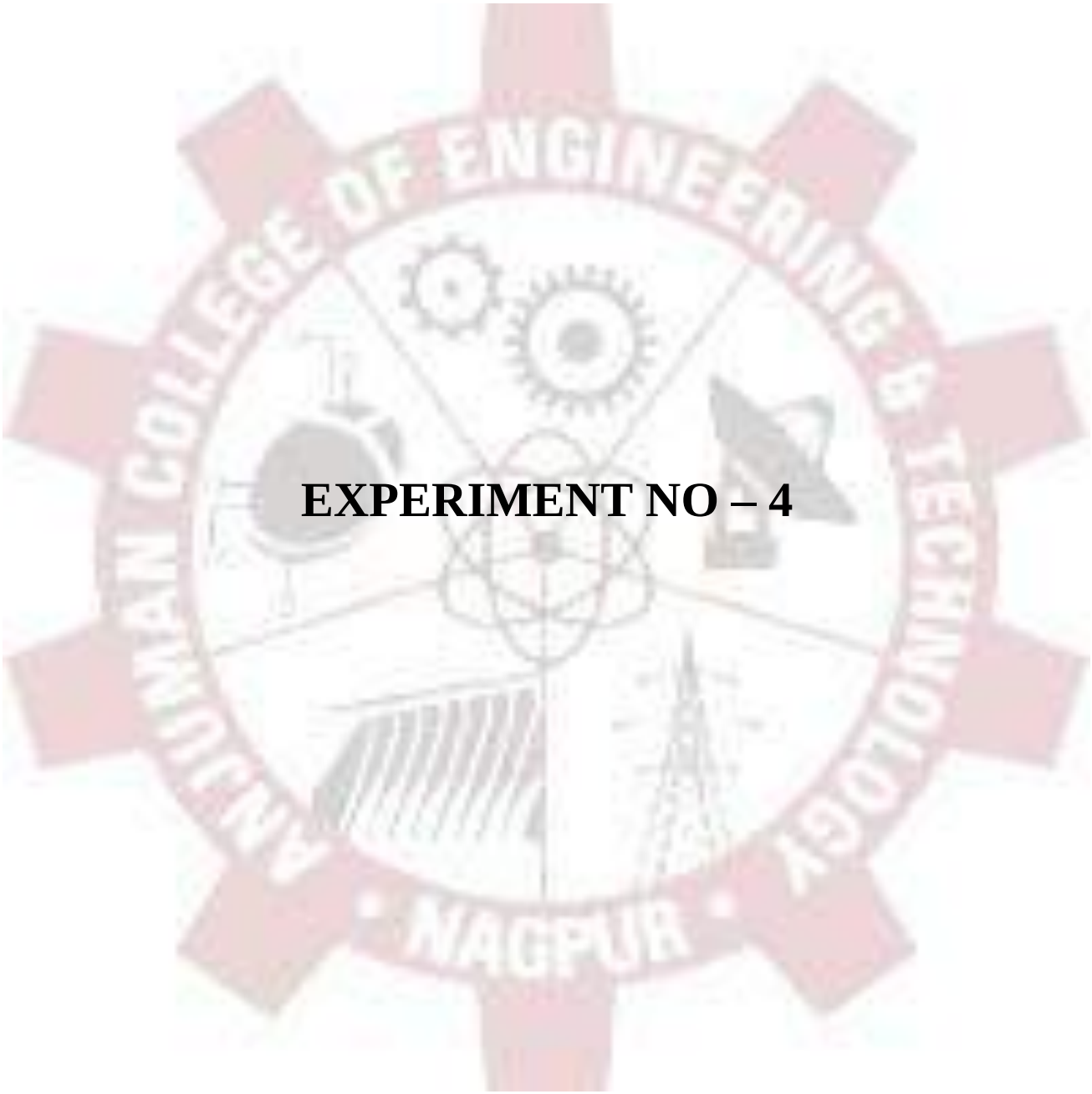
1. What is ArrayStore Exception in java? When you will get this exception?

2. Can you pass the negative number as an array size?

3. Can you change the size of the array once you define it? OR Can you insert or delete the elements after creating an array?

4. What is the difference between int[] a and int a[] ?

Signature of Subject Teacher



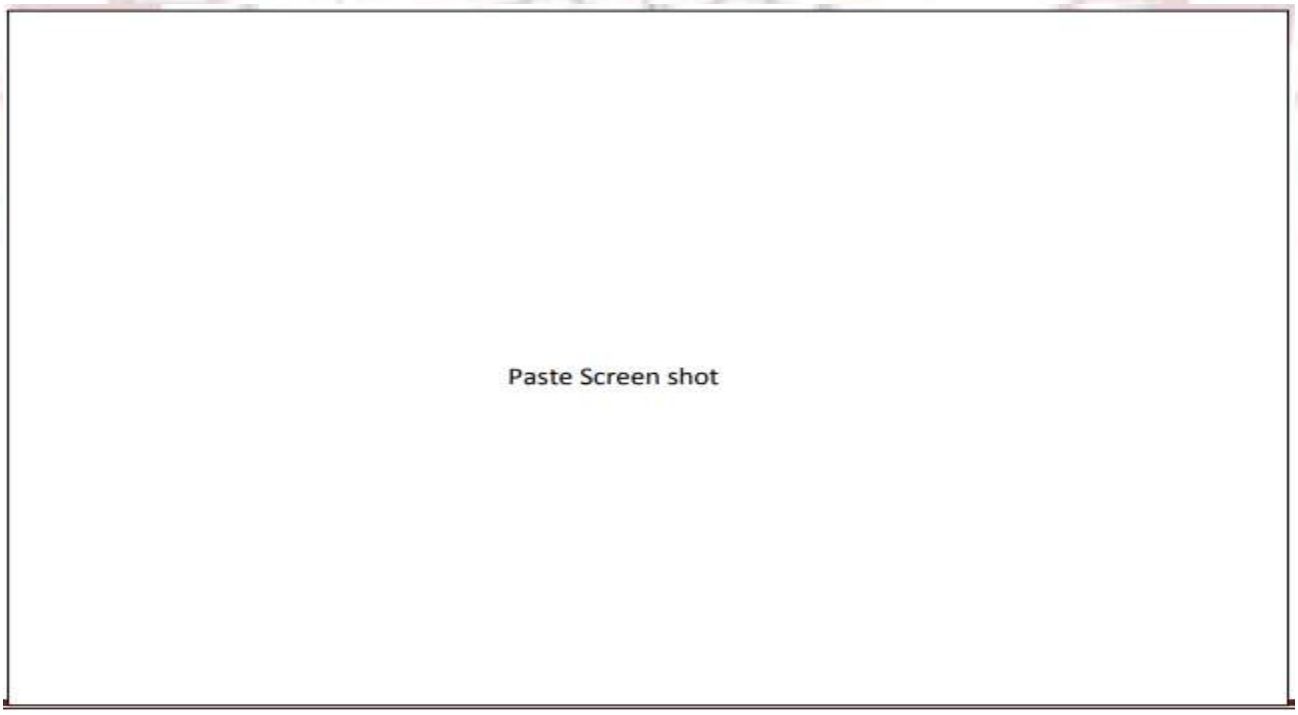
EXPERIMENT NO – 4

Aim: Write a Java Program to convert all primitives into its corresponding wrapper objects and vice-versa.

Program:

```
public class WrapperExample3
{
public static void main(String args[])
{
byte b=10;
short s=20;
int i=30;
long l=40;
float f=50.0F;
double d=60.0D;
char c='a';
boolean b2=true;

//Autoboxing: Converting primitives into objects
Byte byteobj=b;
Short shortobj=s;
Integer intobj=i;
Long longobj=l;
Float floatobj=f;
Double doubleobj=d;
Character charobj=c;
Boolean boolobj=b2;

//Printing objects
System.out.println("---Printing object values---");
System.out.println("Byte object: "+byteobj);
System.out.println("Short object: "+shortobj);
System.out.println("Integer object: "+intobj);
System.out.println("Long object: "+longobj);
System.out.println("Float object: "+floatobj);
System.out.println("Double object: "+doubleobj);
System.out.println("Character object: "+charobj);
System.out.println("Boolean object: "+boolobj);

//Unboxing: Converting Objects to Primitives
byte bytevalue=byteobj;
```

```
short shortvalue=shortobj;
int intvalue=intobj;
long longvalue=longobj;
float floatvalue=floatobj;
double doublevalue=doubleobj;
char charvalue=charobj;
boolean boolvalue=boolobj;

//Printing primitives
System.out.println("---Printing primitive values---");
System.out.println("byte value: "+bytevalue);
System.out.println("short value: "+shortvalue);
System.out.println("int value: "+intvalue);
System.out.println("long value: "+longvalue);
System.out.println("float value: "+floatvalue);
System.out.println("double value: "+doublevalue);
System.out.println("char value: "+charvalue);
System.out.println("boolean value: "+boolvalue);
}}
```



4 b) Creating the Custom Wrapper Class.

Wrapper classes in Java

The **wrapper class in Java** provides the mechanism *to convert primitive into object and object into primitive.*

Since J2SE 5.0, **autoboxing** and **unboxing** feature convert primitives into objects and objects into primitives automatically. The automatic conversion of primitive into an object is known as autoboxing and vice-versa unboxing.

Use of Wrapper classes in Java

Java is an object-oriented programming language, so we need to deal with objects many times like in Collections, Serialization, Synchronization, etc. Let us see the different scenarios, where we need to use the wrapper classes.

- **Change the value in Method:** Java supports only call by value. So, if we pass a primitive value, it will not change the original value. But, if we convert the primitive value in an object, it will change the original value.
- **Serialization:** We need to convert the objects into streams to perform the serialization. If we have a primitive value, we can convert it in objects through the wrapper classes.
- **Synchronization:** Java synchronization works with objects in Multithreading.
- **java.util package:** The java.util package provides the utility classes to deal with objects.
- **Collection Framework:** Java collection framework works with objects only. All classes of the collection framework (ArrayList, LinkedList, Vector, HashSet, LinkedHashSet, TreeSet, PriorityQueue, ArrayDeque, etc.) deal with objects only.

The eight classes of the *java.lang* package are known as wrapper classes in Java.

Autoboxing

The automatic conversion of primitive data type into its corresponding wrapper class is known as autoboxing, for example, byte to Byte, char to Character, int to Integer, long to Long, float to Float, boolean to Boolean, double to Double, and short to Short.

Since Java 5, we do not need to use the valueOf() method of wrapper classes to convert the primitive into objects.

Unboxing

The automatic conversion of wrapper type into its corresponding primitive type is known as unboxing. It is the reverse process of autoboxing. Since Java 5, we do not need to use the intValue() method of wrapper classes to convert the wrapper type into primitives.

Program:

```
class Javacustom
{
private int i;
Javacustom(){}
Javacustom(int i){
this.i=i;
}
public int getValue(){
return i;
}
public void setValue(int i){
this.i=i;
}
@Override
public String toString() {
  return Integer.toString(i);
}
}
//Testing the custom wrapper class
public class TestJavatcustom{
public static void main(String[] args){
Javacstom j=new Javacustom(10);
System.out.println(j);
}
}
```

Paste Screen shot

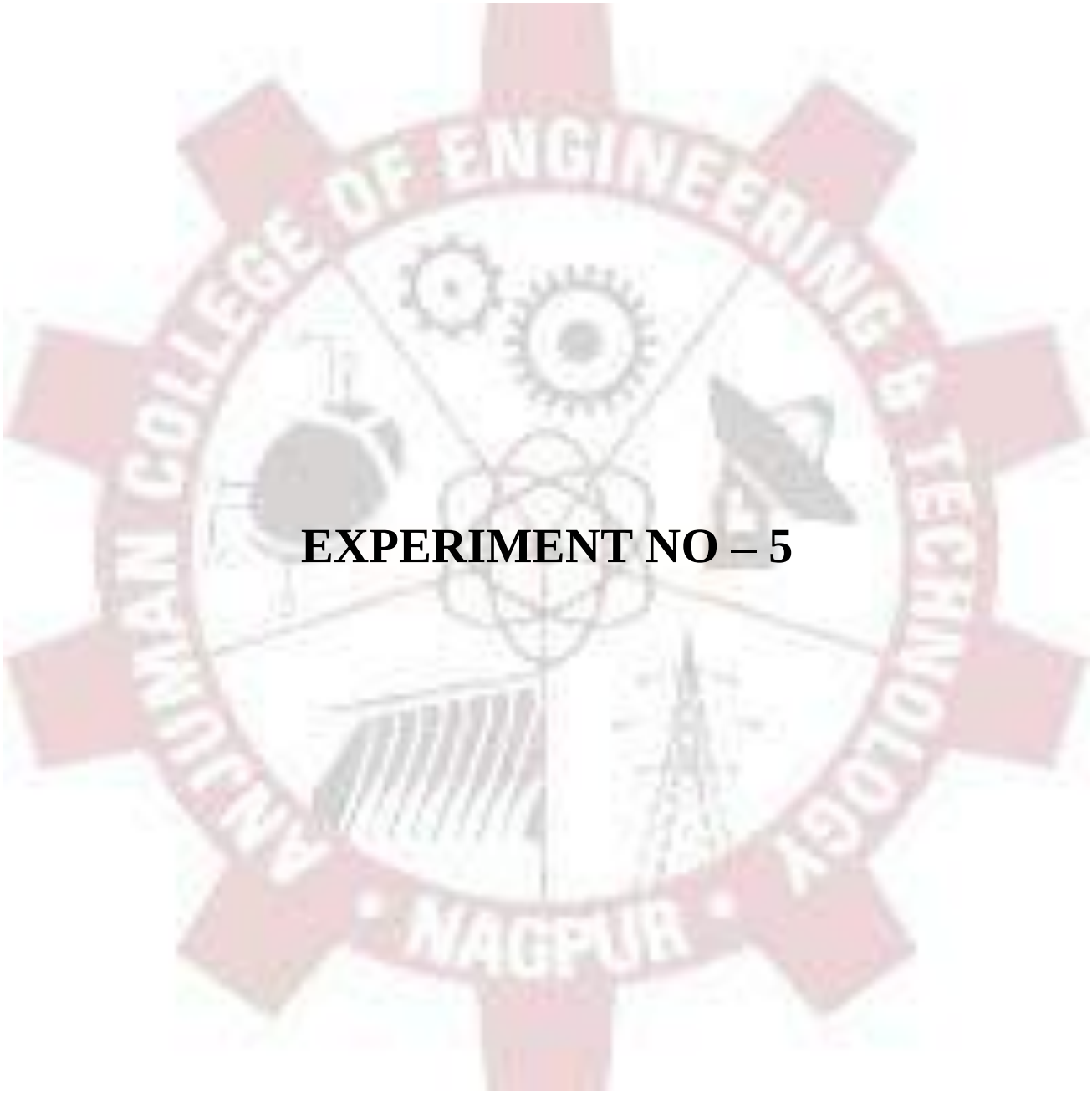
Viva Voce Question

1. What is a Wrapper classes in Java?

2. What are the Uses of Wrapper classes in Java?

3. What is Autoboxing?

Signature of Subject Teacher



EXPERIMENT NO – 5

Aim:- 5 a) Java program to illustrate the concept of Multilevel inheritance.

Program:

```
importjava.lang.*;
importjava.io.*;
class one
{
Public void print_geek()
{
System.out.println("Geeks");
}
}

Class two extends one
{
Public void print_for()
{
System.out.println("for");
}
}

Class three extends two
{
Public void print_geek()
{
System.out.println("Geeks");
}
}

// Drived class public class Main
{
Public static void main(String[] args)
{
three g = new three();
g.print_geek();
 g.print_for();
g.print_geek();
}}
```

Paste Screen shot

5b) A Simple Java program to demonstrate Method Overriding in Java.

Program:

```
// Base Class classParent {
Void show()
{
System.out.println("Parent's show()");
}
}

// Inherited class classChild extends Parent {
// This method overrides show() of Parent @Override
Void show()
{
System.out.println("Child's show()");
}
}

// Driver class classMain {
Public static void main(String[] args)
{
// If a Parent type reference refers
// to a Parent object, then Parent's
// show is called
Parent obj1 = new Parent();
 obj1.show();

// If a Parent type reference refers
// to a Child object Child's show()
// is called. This is called RUN TIME
// POLYMORPHISM.
Parent obj2 = new Child();
obj2.show();
}}
```

Paste Screen shot

5 c) Write a java program in which you will declare an abstract class Vehicle, inherits this class from two classes car and truck using the method engine in both. Display "car has good engine" and "truck has bad engine".

Program:

```
abstract class vechile
{
abstract void engine();
}
class car extends vechile
{
public void engine()
{
System.out.println("car has good engine");
}}
class truck extends vechile
{
public void engine()
{
System.out.println("truck has bad engine");
}}
public class TestVechile
{
public static void main(String arr[])
{
vechile v=new car();
v.engine();
vechile n=new truck();
n.engine();
}}
```

Paste Screen shot

Viva Voce Question

1. What is Inheritance in Java?

2. How to use Inheritance in Java?

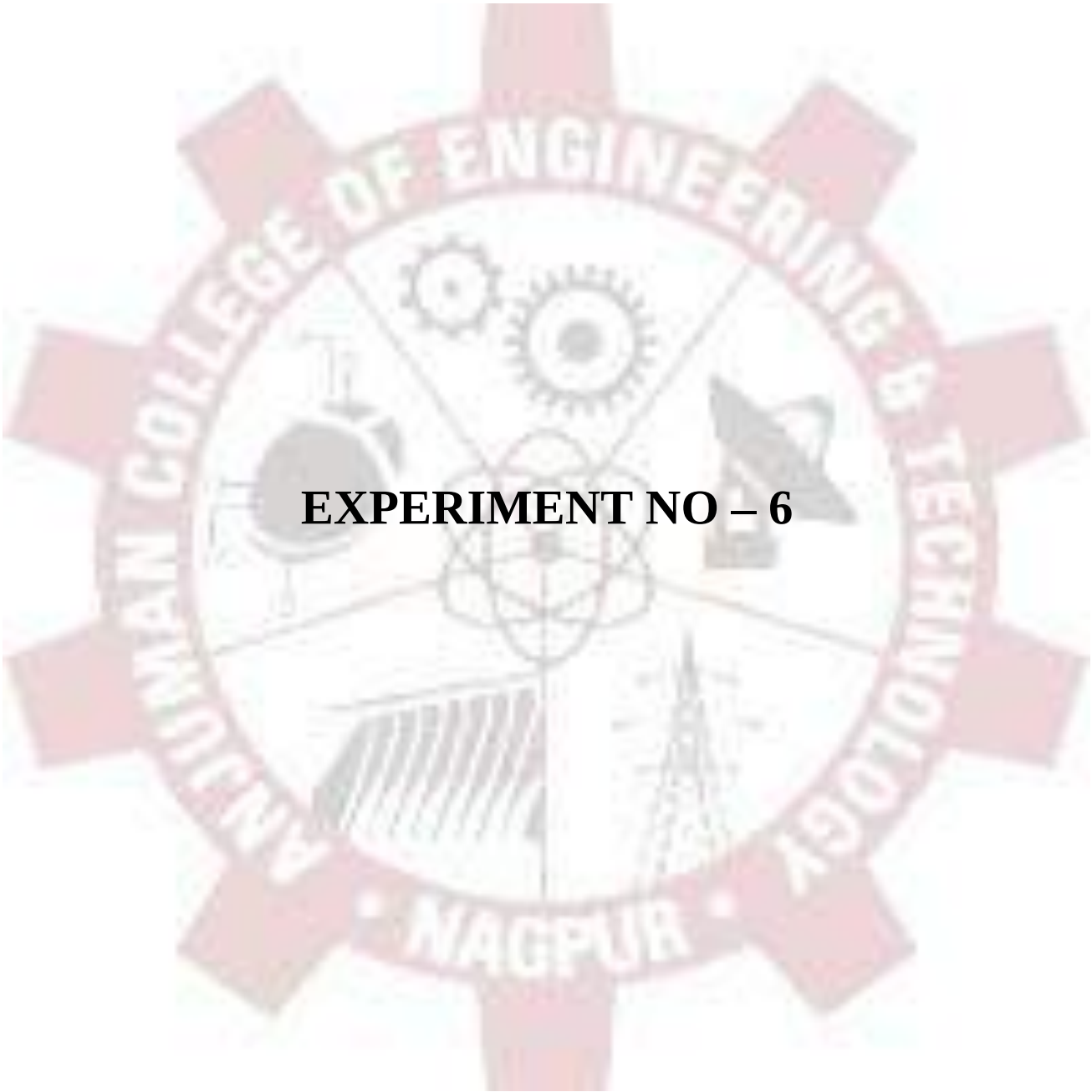
3. Can A Class Extend More Than One Class In?

4. Is it compulsory for a class which is declared as abstract to have at least one abstract method?

5. Abstract class must have only abstract methods. True or false?

6. What is the main difference between abstract method and final method?

Signature of Subject Teacher



EXPERIMENT NO – 6

Aim:- 6 a) Write a program in java if number is less than 10 and greater than 50 it generate the exception out of range. Else it displays the square of number.

Program:

```
Class CustomTest
{
public static void main(String arr[])
{
try
{
int a=Integer.parseInt(arr[0]);
if(a<0|| a>50)
throw(new outofRangeException("valid range is 10 to 50"));
{
int s=a*a;
System.out.println("Square is:"+s);
}
}
catch(Exception ex)
{
System.out.println(ex);
}
}
}
```

Paste Screen shot

6b) Write a program in java to enter the number through Command line argument, if first and second number is not entered it will generate the exception. Also divide the first number with second number and generate the arithmetic exception.

Program:

```
class Divide2
{
public static void main(String arr[])
{
try
{
if(arr.length<2)

throw(new Exception("two argument must be provided"));
int a= Integer.parseInt(arr[0]);
int b=Integer.parseInt(arr[1]);
if(b==0)
throw(new Exception("second argument should be non zero"));
int c=a/b;
System.out.println("result:"+c);
}
catch(Exception e)
{
System.out.println(e);
}
}
}
```

Paste Screen shot

6c) Write a program in Java to call the method divide, the first number with second and generate the exception.

Program:

```
class Divide3
{
public static int divide(int x, int y)
{
int z=0;
try
{
try
{
z= x/y;
}
finally
{
//return Z;
}
}
catch(ArithmeticException ex)
{
System.out.println(ex);
}
return z;
}
public static void main(String arr[])
{
try
{
int a=Integer.parseInt(arr[0]);
int b=Integer.parseInt(arr[1]);
int c=divide(a,b);
System.out.println("Result is="+c);
}
catch(Exception e)
{
System.out.println(e);
}
}
}
```

Paste Screen shot

Viva Voce Question

1. What is Exception in Java?

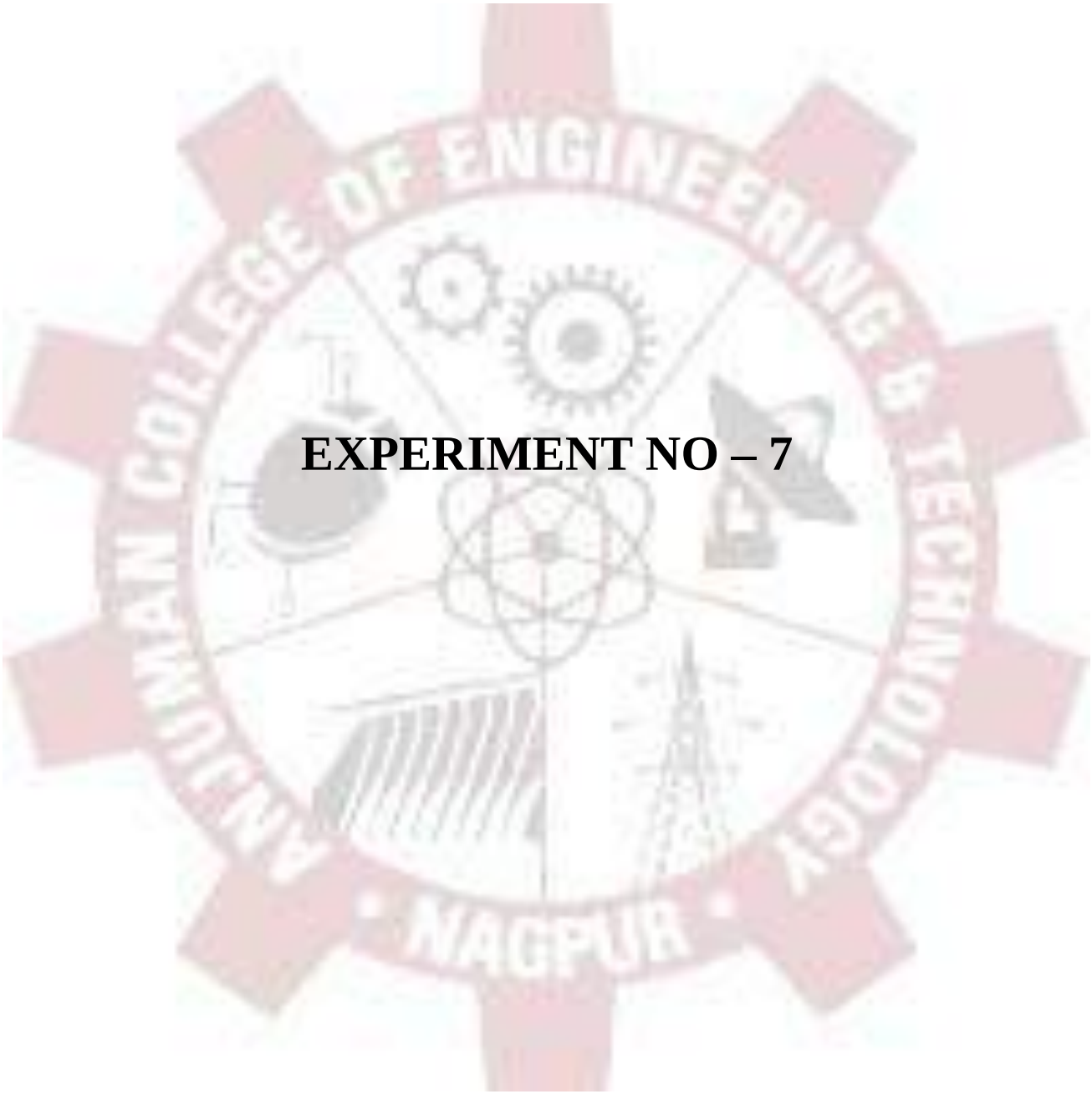
2. What are the Exception Handling Keywords in Java?

3. What is difference between throw and throws keyword in Java?

4. What happens when exception is thrown by main method?

5. What is difference between Checked and Unchecked Exception in Java?

Signature of Subject Teacher



EXPERIMENT NO – 7

7a) Write a java program in which thread sleep for 5 sec and change the name of thread.

Program:

```
Import java.lang.*;
Class ThreadTest extends Thread
{
static
{
Thread t = Thread.currentThread();

//Thread t=new Thread. currentThread();
System.out.println("thread test is loaded by"+t.getName()+"thread");
t.setName("vishal");
System.out.println("changed the name of thread");
System.out.println("suspending thread for 5 sec");
try
{
Thread.sleep(5000);
}
catch(Exception ex){}
}
public static void main(String arr[])
{
Thread t=Thread.currentThread();
System.out.println("main() is invoked in"+t.getName()+"thread...");
}
}
```

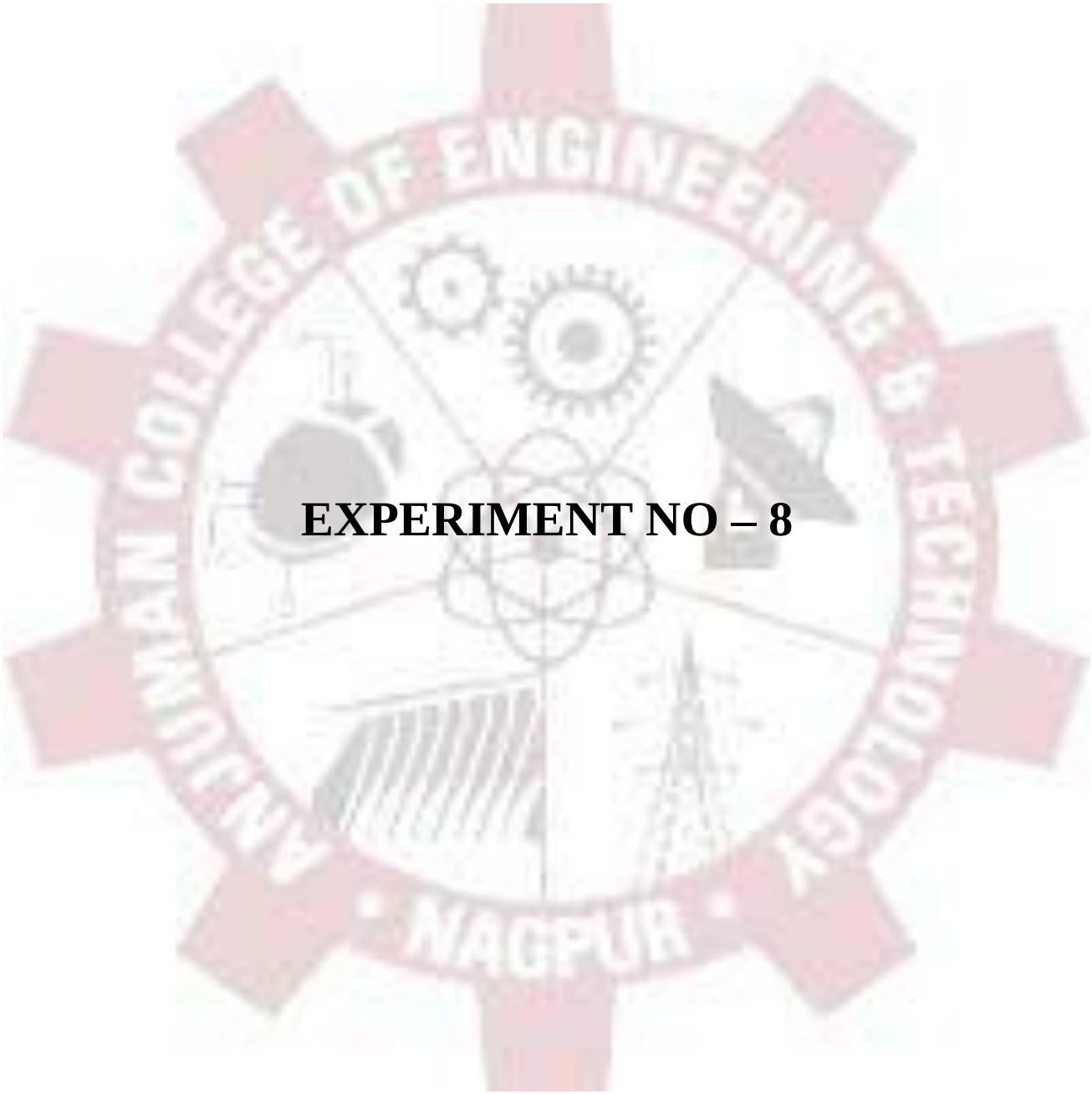
Paste Screen shot

7 b) Write a java program in which thread sleep for 6 sec in the loop in reverse order from 5 to 1 and change the name of thread.

Program:

```
importjava.lang.*;
class Thread1
{
public static void main(String arr[])
{
Thread t=Thread. currentThread();
 System.out.println("current thread is:"+t);
 t.setName("vishal thread");
System.out.println("after name chage thread:"+t);
try
{
for(int n=5; n>0; n--)
{
System.out.println(n);
Thread.sleep(6000);
}
}
catch(InterruptedException e)
{
System.out.println("main thread is interrupted");
}
}
}
```

Paste Screen shot



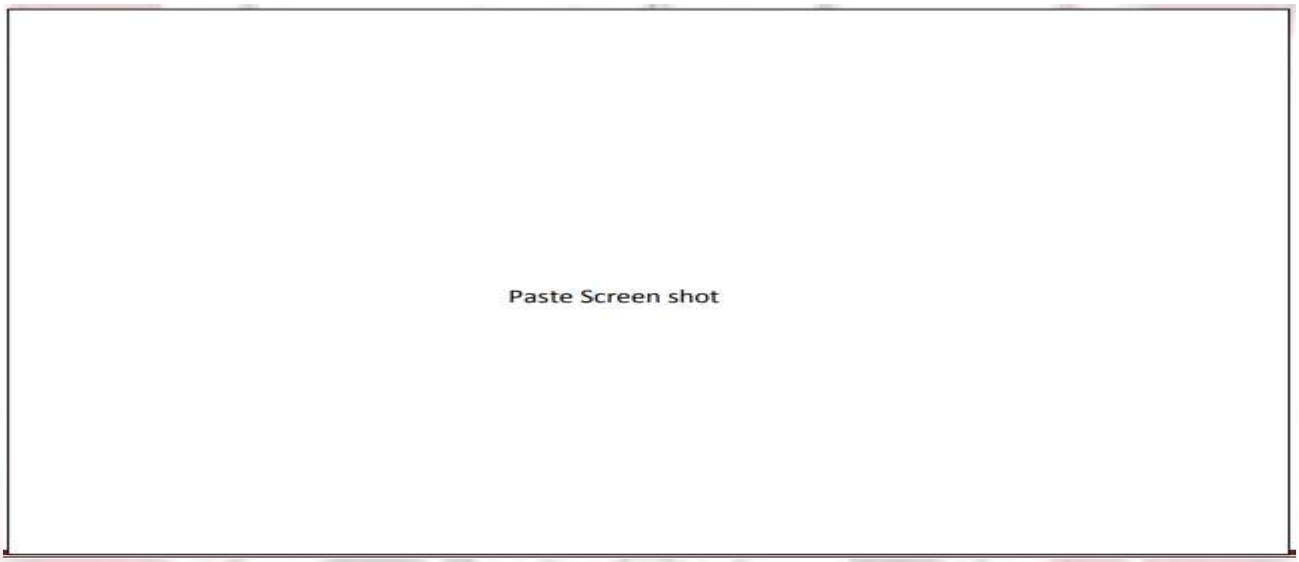
EXPERIMENT NO – 8

8a) Write a java program for multithread in which user thread and thread started from main method invoked at a time each thread sleep for 1 sec.

Program:

```
Class UserThread extends Thread
{
public void run()
{
Thread t=Thread. currentThread();
System.out.println("run() is invoked in"+t.getName()+"thread...");
for(int i=1;i<=10;i++)
{
System.out.println("run:"+i);
 try
{
Thread.sleep(1000);
}
catch(Exception e)
{
}
}
System.out.println("run() is completed");
}
}
Class MultiThread
{
public static void main(String arr[])
{
System.out.println("main() started creating an object of user Thread.          ");
UserThread t=new UserThread();
System.out.println("directly invoking run() of user thread");
t.run();
System.out.println("control back in main()    ");
System.out.println("launching new thread for run() of user thread.   ");
t.start();
for(int i=10;i>0;i--)
{
System.out.println("main:"+i);
try
{
Thread.sleep(1000);
}
catch(Exception e)
{
}
```

```
}
System.out.println("main() completed");
}
}
```



8b)Write a java program for to solve producer consumer problem in which a producer produce a value and consumer consume the value before producer generate the next value.

Program:

```
class Buffer
{
int value;
boolean produced=false;
public synchronized void produce(int x)
{
if(produced)
{
System.out.println("producer enter monitor out of turn..suspend.     ");
try
{
wait();
}
catch(Exception e)
{
}
}
value=x;
System.out.println(value+"is produced");
produced=true;
notify();
}
public synchronized void consume()
{
if(! produced)
{
System.out.println("consumer entered the monitor out of turn,suspend.       ");
try
{
wait();
}
catch(Exception e)
{}
}
System.out.println(value+"is consumed");
produced=false;
notify();
}
```

```
}
class Producer extends Thread
{
Buffer buffer;
public Producer(Buffer b)
{
buffer =b;
}
public void run()
{
System.out.println("producer started ,producing value.        ");
for(int i=1;1<=10;i++)
buffer.produce(i);
}
}
class Consumer extends Thread
{
Buffer buffer;
public Consumer(Buffer b)
{
buffer =b;
}
public void run()
{
System.out.println("consumer started,consuming value.       ");
for(int i=1;i<=10;i++)
buffer.consume();
}
}
class PC1
{
public static void main(String arr[])
{
Buffer b=new Buffer();
Producer p=new Producer(b);
Consumer c=new Consumer(b);
 p.start();
c.start();
}
}
```

Paste Screen shot

Viva Voce Question

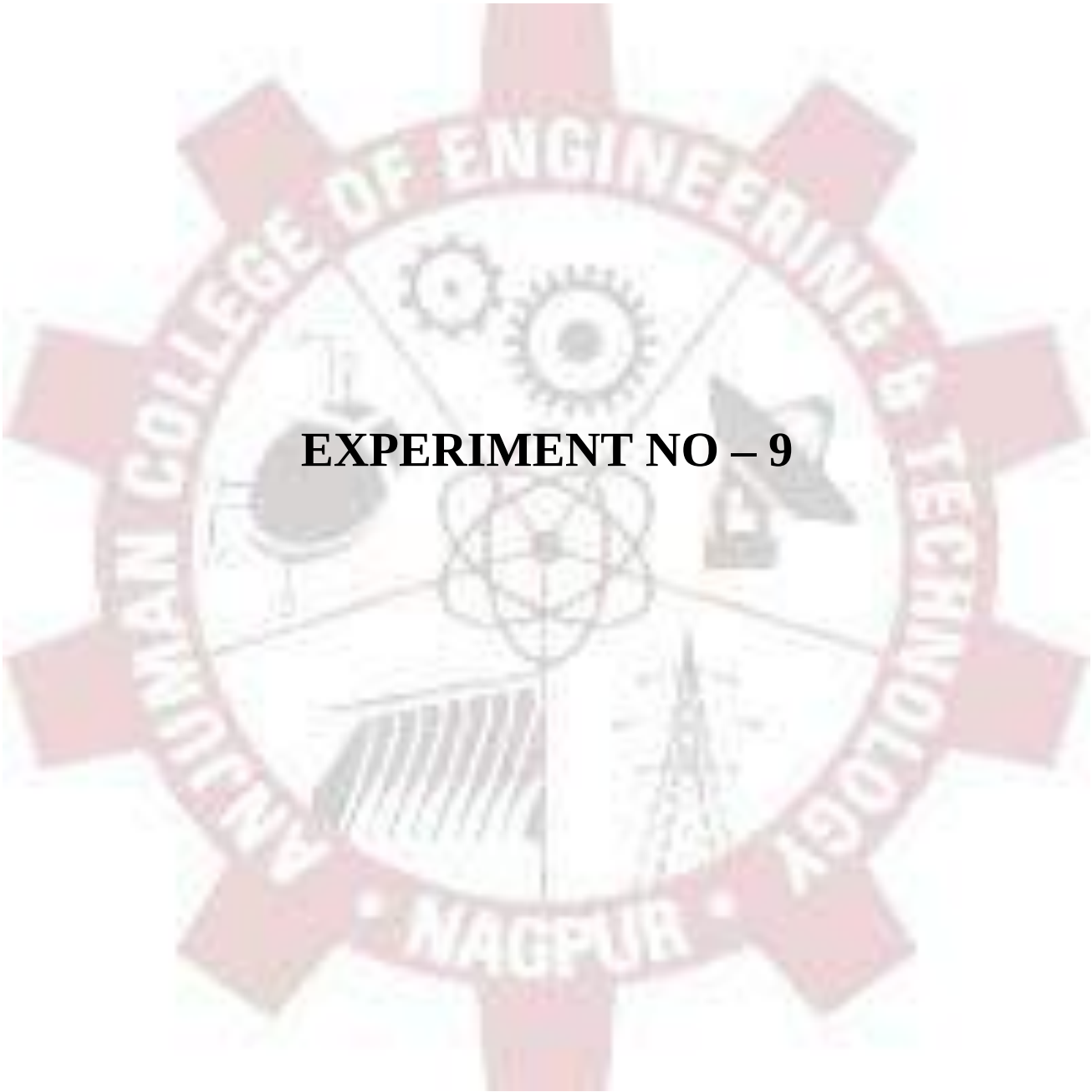
1. What is thread in Java?

2. What is Multithreading?

3. What are the two ways of creating a thread?

4. Can we call run() method of Thread class?

Signature of Subject Teacher



EXPERIMENT NO – 9

Aim: Java program on Array list.

Program:

```
public class ArrayListTest
 {
   public static void main(String[] args)
{
    ArrayList al = new ArrayList();
    al.add("aaa");
    al.add("bbb");
    al.add("ccc");
    // see return type of add below
    System.out.println(al.add("ddd"));
    al.size();
    System.out.println(al.size);
    //to check if Arraylist is empty
    al.isEmpty();
  System.out.println("iteration of Arraylist by for loop");
  for (int i = 0; i < al.size(); i++)
{
     System.out.println(al.get(i));
   }
  System.out.println("iteration of Arraylist by Iterator");
  Iterator itr = al.iterator();
  while (itr.hasNext())
  {
        Object o = itr.next(); //this is removed in
                               //jdk 1.5 and after by autoboxing
        String s = (String) o;
        System.out.println(s);
       }
  System.out.println("iteration of Arraylist by List Iterator");
  ListIterator ltr = al.listIterator();
  while (ltr.hasNext())
  {
        Object o = ltr.next();
        String s = (String) o;
          System.out.println(s);
        //Object op = ltr.previous();
        /*String prevStr = (String) op;
        System.out.println(prevStr);
      }
   }
}
```

Paste Screen shot

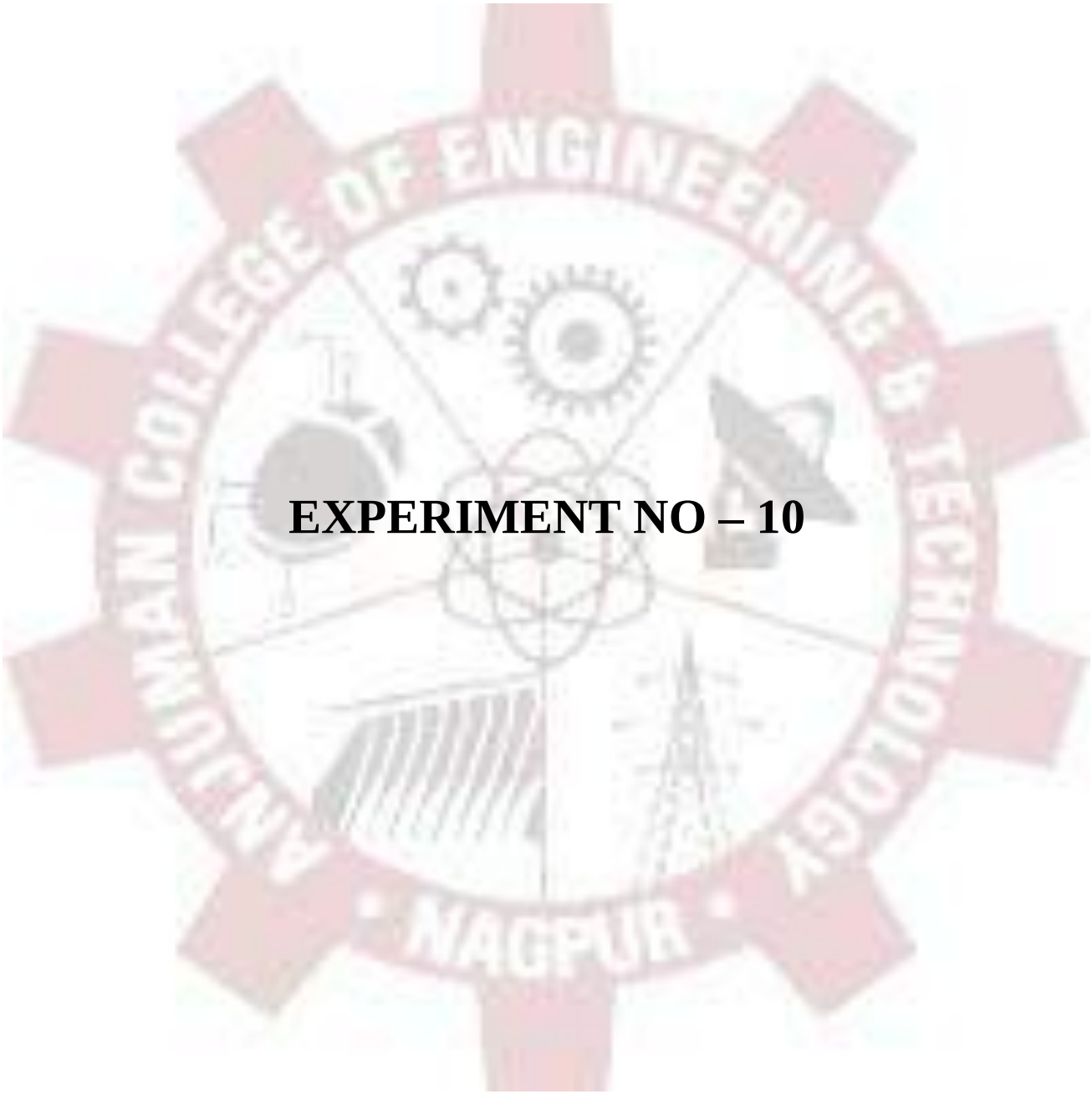
Viva Voce Question

1. What is List Explain?

2. What is Array List?

3. What is Vector Class?

Signature of Subject Teacher



EXPERIMENT NO – 10

Aim: Program to demonstrate HashMap.

Program:

```
public class HashMapTest
{
 public static void main(String[] args)
{
    // Creation of HashMap
        HashMap hm = new HashMap();
    // Adding elements with key
        hm.put("101", "java");
        hm.put("102", ".Net");
    // Will print null
        Object o = hm.put("103", "C++");
        System.out.println(o);
    // Will print previous value as it is duplicate value
        Object o1 = hm.put("103", "C");
        System.out.println(o1);

//1. Retrieving elements from HashMap by using iterator
   System.out.println("======By using Iterator======");
   Set s = hm.keySet(); // set s contains all keys
    Iterator itr = s.iterator();
    while (itr.hasNext())
 {
    String key = (String) itr.next();
    System.out.println("Key :"+key);
    System. out. println(" Value :"+ hm.get(key));
  }

//2. Retrieving elements from HashMap by using Map.Entry

   System.out.println("====== By using Map.Entry ======");
   // Get a set of the entries
   Set set = hm.entrySet();
   // Get an iterator
   Iterator it = set.iterator();

   // Display elements
   while(it.hasNext())
{
        Map.Entry me = (Map.Entry) it.next();
        System.out.print(me.getKey()+ ": ");
        System.out.println(me.getValue());
      }
   }
}
```

Paste Screen shot

Viva Voce Question

1. What is Map in Java?

2. What is HashMap?

3. What is Linked HashMap?

Signature of Subject Teacher