SETHU INSTITUTE OF TECHNOLOGY

An Autonomous Institution | Accredited By NAAC with 'A++' Grade

Pulloor, Kariapatti –Taulk. Virudhunagar Dist-626115.

Department of Artificial Intelligence and Data Science

CYBER SECURITY – 21CSV604

Assignment – I

Done by:

The Pegasus Spyware Scandal: Surveillance, Ethics, and Mobile Security in the Digital Age

---

Introduction

In an era increasingly defined by digital connectivity, concerns over digital surveillance and privacy have come to the forefront of global discourse. The Pegasus spyware scandal, a revelation that shook governments and civil societies alike, highlighted the vulnerabilities of even the most secure mobile devices. Developed by the NSO Group, Pegasus was initially promoted as a tool to aid governments in combating terrorism and organized crime. However, revelations exposed its deployment against journalists, human rights activists, and political opponents, igniting widespread concern over the ethical and legal use of surveillance technology.

This report aims to:

- Explore the origins and deployment of Pegasus spyware.

- Examine real-world abuses and their impact.

- Understand the mechanisms of mobile surveillance.

- Discuss the legal and ethical frameworks surrounding digital forensics.

- Offer lessons learned and recommendations for the future.

What Pegasus spyware can do

Source: Pegasus Project — BBC

Background and Context

Pegasus Spyware: Origins and Development

The Pegasus spyware was developed by the Israeli cybersecurity firm NSO Group, founded in 2010. Marketed to law enforcement and intelligence agencies, Pegasus was designed to provide "lawful surveillance" capabilities to track criminals and terrorists. However, its powerful and covert capabilities soon attracted scrutiny as reports emerged of its misuse.

Intended vs. Real-World Use: While NSO Group claimed to restrict its clients to government agencies with good human rights records, investigations revealed Pegasus was used to surveil journalists, activists, lawyers, and opposition figures in countries with questionable human rights practices.

The Scandal Unfolds

Timeline of Revelations:

- **2016:** First discovery of Pegasus on an activist's iPhone by Citizen Lab.

- **2018-2019:** Reports of its use against journalists and civil society members in Mexico and the Middle East.

- **July 2021:** The Pegasus Project, a global investigative collaboration, revealed a list of over 50,000 phone numbers potentially targeted by Pegasus.

Key Actors Involved:

- Targeted individuals included journalists from The Guardian and Le Monde, Saudi dissident Jamal Khashoggi's associates, and Indian opposition politicians.

- Governments of Hungary, India, Mexico, Saudi Arabia, and the UAE were implicated.

Surveillance and Spyware Abuse

The Mechanics of Surveillance

Pegasus is capable of infecting smartphones using various vectors, including spear-phishing messages and more alarmingly, zero-click exploits that require no user interaction.

Surveillance Operations: Government agencies may deploy spyware through covert operations, often justifying their use under the guise of national security. However, non-state actors such as private contractors have also played roles in surveillance activities.

Spyware Abuse: Case Studies

Case Study 1: Journalists in Mexico Over two dozen Mexican journalists were found to be targeted by Pegasus, especially those investigating corruption and drug cartels.

Case Study 2: Jamal Khashoggi Associates of the slain Saudi journalist Jamal Khashoggi were allegedly surveilled by Pegasus both before and after his assassination.

Impact on Civil Liberties:

- Breach of freedom of expression.

- Psychological distress and mistrust within civil societies.

- Erosion of trust in government accountability.

Key Concepts in Mobile Device Security

Understanding Mobile Security Threats

Modern mobile devices face a range of threats, including:

- Malware and spyware.

- Phishing attacks.

- Exploitation of unpatched software vulnerabilities.

Pegasus exemplifies these risks, showcasing how even up-to-date phones can be compromised silently.

Zero-Click Exploits

Definition and Importance: Zero-click exploits allow spyware to infiltrate devices without user interaction, typically via messaging apps like iMessage and WhatsApp.

Pegasus Use of Zero-Click:

- iOS vulnerabilities were exploited to silently install Pegasus.

- The 2019 WhatsApp vulnerability enabled Pegasus to infect over 1,400 devices globally.

Detection and Mitigation Challenges:

- Encrypted messaging platforms and advanced obfuscation make detection extremely difficult.

- Most antivirus tools cannot detect Pegasus.

Ethical Considerations in Cyber Forensics

Legal and Ethical Frameworks

Relevant laws include:

- The Budapest Convention on Cybercrime.

- National regulations like the IT Act in India and the Patriot Act in the U.S.

Dilemmas in Investigation:

- Identifying state-backed actors is technically and politically complex.

- Attribution errors may damage reputations or cause international tension.

Ethical Dilemmas: Case Discussions

Balancing Privacy vs. Security: While surveillance can protect citizens, it also risks violating fundamental rights if unchecked.

Responsibilities of Cybersecurity Experts:

- Ensure transparency and accountability.

- Avoid conflicts of interest when working with governments or private clients.

Consequences of Misuse:

- Legal liability for unethical surveillance.

- Loss of public trust in technology.

Lessons Learned and Future Directions

Impacts on Policy and Technology

Policy Reactions:

- Several governments launched inquiries or legal actions.

- The European Parliament initiated a special committee to investigate Pegasus abuse.

Technological Improvements:

- Apple and Google have released updates to close known vulnerabilities.

- Introduction of "Lockdown Mode" in iOS to mitigate zero-click attacks.

Recommendations

For Users:

- Regularly update devices.

- Use end-to-end encrypted communication tools.

- Enable features like Lockdown Mode.

For Policymakers:

- Establish independent oversight for surveillance technology use.

- Ban export of spyware to regimes with poor human rights records.

For Developers and Security Researchers:

- Invest in threat detection tools.

- Conduct vulnerability bounty programs to identify zero-day flaws.

Conclusion

The Pegasus spyware scandal serves as a stark reminder of the dual-edged nature of surveillance technologies. While they can provide national security benefits, their unchecked use poses grave threats to human rights and democracy. Moving forward, the global community must remain

vigilant, fostering transparent governance, advancing digital security, and upholding the fundamental right to privacy.

References

1. "The Pegasus Project: Exposing the Spyware Scandal," *The Guardian*.

2. Marczak, B., Scott-Railton, J., et al. (2021). "Zero-Click Exploits: How Pegasus Works," *Citizen Lab*.

3. "After Pegasus: Mobile Security and Policy Responses," *The Wire*.

4. Amnesty International, "Forensic Methodology Report: How Amnesty Technical Lab Uncovered the Pegasus Project."

5. European Parliament Special Committee on Pegasus and Surveillance Technologies (2022).

SETHU INSTITUTE OF TECHNOLOGY

An Autonomous Institution | Accredited By NAAC with 'A++' Grade

Pulloor, Kariapatti –Taulk. Virudhunagar Dist-626115.

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

LAB MANUAL & RECORD

YEAR – III

SEMESTER – VI

21UAD603 – THINKING IN JAVA (INTEGRATED)

NAME   : _____

ROLL NO   : _____

REG. NO   : _____

DEPARTMENT   : _____

SETHU INSTITUTE OF TECHNOLOGY

PULLOOR - 625 115. KARIAPATTI TALUK, VIRUDHUNAGAR DISTRICT

PRACTICAL RECORD

BONAFIDE CERTIFICATE

Certified that this the bonafide record of work done by

....

in the ....

Laboratory during the year.......................................

Staff-in-charge   Head of the Department

Reg. No. :

Submitted for the Practical Examination held on ....

Internal Examiner   External Examiner

INSTITUTE VISION AND MISSION

INSTITUTE VISION
- To promote excellence in technical education and scientific research for the benefit of the Society

INSTITUTE MISSION
- To provide Quality Technical Education to fulfill the aspiration of the students and to meet the needs of the industry.
- To provide holistic learning ambience.
- To impart skills leading to employability and entrepreneurship.
- To establish effective linkage with industries.
- To promote Research and Development Activities.
- To offer Services for the Development of Society through education and technology.

Core Values
- Quality
- Commitment
- Innovation
- Team work
- Courtesy

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

QUALITY POLICY
- To provide Quality technical education to the students
- To produce competent professionals and contributing citizens
- To contribute for the upliftment of the society

DEPARTMENT VISION
- To produce high quality technologists for the dynamic societal needs in the field of Artificial Intelligence and Data Science

DEPARTMENT MISSION
- Providing quality education and support innovation in expert systems and data science to meet industry expectations.
- Offering holistic learning ambience.
- Developing the skills of the students to make successful engineers and entrepreneurs.
- Creating relationship with the industries for mutual knowledge transfer.
- Encouraging Research activities related to industry and society.

PROGRAM EDUCATIONAL OBJECTIVES

PEO 1	Graduates will succeed as successful engineers in the field of Artificial Intelligence and Data Science for pursuing inter disciplinary projects for the development of the nation.[**Core Competence**]
PEO 2	Graduates will work as team leaders and members with professional behavior and ethics.[**Professionalism**]
PEO 3	Graduates will enrich their professional skills through higher studies, employability, and research activities for the benefit of the society.[**Life-Long Learning**]

PROGRAM SPECIFIC OUTCOMES

PSO – 1	Interpret data, use software tools to conduct experiments, and apply AI & machine learning techniques to solve multi-disciplinary problems.
PSO – 2	Apply standard practices, strategies and use appropriate models of data analytics to discover knowledge.

PROGRAM OUTCOMES

1.	Apply the knowledge of mathematics, basic sciences, engineering fundamentals, and Computer Science and Design to the solution of complex engineering problems. **(Engineering Knowledge)**
2.	Identify, formulate, review research literature and analyze complex engineering problems requiring computing solutions to reach substantiated conclusions using first principles of mathematics, basic sciences, and Computer Science and Design. (Problem analysis)
3.	Design solutions for computer applied 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. (Design/development of solutions)
4.	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. (Conduct investigations of complex problems)
5.	Create, Select and apply appropriate techniques, resources, and modern IT tools including prediction and modeling to computing related complex engineering activities with an understanding of the limitations. (Modern tool usage)
6.	Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional computer science and engineering practice. (The Engineer and society)
7.	Understand the impact of the professional computer science and design solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development. (Environment and sustainability)
8.	Apply ethical principles and commit to professional ethics and responsibilities and norms of the computer science and design practice. (Ethics)
9.	Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings. (Individual and team work)
10.	Communicate effectively on complex computer science and design 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. (Communication)
11.	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 cost effective projects in multidisciplinary environments.(Project management and finance)
12.	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. (Life-long learning)

Course Code	Course Name	L	T	P	C
21UAD603	Thinking in JAVA (Integrated)	0	0	2	1

COURSE OBJECTIVE:

- To impart object oriented programming principles to students
- To demonstrate java programming language principles to develop application programs.
- To enable students to create GUI based applications in Java programming language.
- To enable students to establish database connectivity from Java.

List of Experiments:

1. Develop a Java application to generate Electricity bill.
2. Implement a program to convert temperature from Fahrenheit to Celsius degree.
3. Develop a java application to implement currency, time and distance converters using packages.
4. Develop a java application with Employee payroll system.
5. Create a java application with the instance variables and methods.
6. Design a Java interface for ADT Stack.
7. Design a Java Application for String Operations using ArrayList.
8. Design a Java Application using getter and setter methods.
9. Write a Java Program using an abstract class.
10. Write a Java program to implement custom user defined exception handling.
11. Write a Java program that reads a file name from the user, displays information.
12. Write a java program that implements a multi-threaded application.
13. Write a java program using a generic max finder.
14. Write a java program using Constructors and Methods.
15. Write a java program using Constructors Overloading.
16. Write a Java program to implement user defined exception handling.
17. Design a Java Application using ArrayList.
18. Design a Java Application using ArrayList getting input and output from user and print.
19. Design a JAVA Application for creating a Calculator Application using packages.
20. Design a Simple Text Editor using Java Applets.

TOTAL: 30 PERIODS

COURSE OUTCOMES

After the successful completion of this course, the student will be able to

CO. No	Course Outcomes	PO Mapping
CO1	Explain the Object oriented features of Java	-
CO2	Write Java code for various applications	PO1,PSO1
CO3	Analyze the suitable object oriented methodology for solving a complex engineering problem	PO2,PSO2
CO4	Compare the given code with original for logical and syntactical errors	PO4
CO5	Design various real time java applications	PO3,PSO2
CO6	Use modern tools to implement coding	PO5

HARDWARE AND SOFTWARE REQUIRMENTS

Computer Required: 60 No's

Minimum Requirement: Processor: Pentium IV, Ram: 4 GB, Hard Disk: 1 TB

SOFTWARE REQUIREMENTS

Windows, JAVA IDE or Equivalent.

ATTAINMENT OF PROGRAM OUTCOMES & PROGRAM SPECIFIC OUTCOMES

Exp. No.	Experiment	PO Attained	PSO Attained
1	Develop a Java application to generate Electricity bill.	PO1, PO2	PSO1
2	Implement a program to convert temperature from Fahrenheit to Celsius degree.	PO1, PO2	PSO1
3	Develop a java application to implement currency, time and distance converters using packages.	PO1, PO2	PSO1
4	Develop a java application with Employee payroll system.	PO1, PO2	PSO1
5	Create a java application with the instance variables and methods.	PO1, PO2	PSO1
6	Design a Java interface for ADT Stack.	PO1, PO2	PSO1
7	Design a Java Application for String Operations using ArrayList.	PO1, PO2	PSO1
8	Design a Java Application using getter and setter methods.	PO1, PO2, PO3	PSO1
9	Write a Java Program using an abstract class.	PO1, PO2	PSO1
10	Write a Java program to implement custom user defined exception handling.	PO1, PO2	PSO1
11	Write a Java program that reads a file name from the user, displays information.	PO1, PO2	PSO1
12	Write a java program that implements a multi-threaded application.	PO1, PO2	PSO1
13	Write a java program using a generic max finder.	PO1, PO2	PSO1
14	Write a java program using Constructors and Methods.	PO1, PO2	PSO1
15	Write a java program using Constructors Overloading.	PO1, PO2	PSO1
16	Write a Java program to implement user defined exception handling.	PO1, PO2	PSO1
17	Design a Java Application using ArrayList.	PO1, PO2	PSO1
18	Design a Java Application using ArrayList getting input and output from user and print.	PO1, PO2	PSO1
19	Design a JAVA Application for creating a Calculator Application using packages.	PO1, PO2, PO3	PSO1
20	Design a Simple Text Editor using Java Applets.	PO1, PO2, PO3	PSO1

COURSE OUTCOME AND PROGRAM OUTCOME MAPPING

Course Outcomes	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 01	PSO 02
Explain the Object oriented features of Java	3	3												3
Write Java code for various applications	3	3	3											3
Analyze the suitable object oriented methodology for solving a complex engineering problem	3	3	3										3	
Compare the given code with original for logical and syntactical errors	3	3	3										3	
Design various real time java applications	3	3	3						3				3	
Use modern tools to implement coding	3	3	3		3					3			3	3
21UAD603	3	3	3		3				3	3			3	3

| Ex.No: 1 | **Electricity Bill Calculator** |
| Date: | |

Aim:

Develop a Java application to generate Electricity bill. Create a class with the following members: Consumer no., consumer name, previous month reading, current month reading, type of EB connection (i.e domestic or commercial). Compute the bill amount using the following tariff.

If the type of the EB connection is domestic, calculate the amount to be paid as follows:

- First 100 units - Rs. 1 per unit
- 101-200 units - Rs. 2.50 per unit
- 201 -500 units - Rs. 4 per unit
- >501 units - Rs. 6 per unit

If the type of the EB connection is commercial, calculate the amount to be paid as follows:

- First 100 units - Rs. 2 per unit
- 101-200 units - Rs. 4.50 per unit
- 201 -500 units - Rs. 6 per unit
- >501 units - Rs. 7 per unit

Algorithm:

Step 1 Start the process

Step 2 Get the user information's [Name, Consumer Number, Reading's of previous and current month,Connection Type]

Step 3 Compute units consumed by user [Current Month Reading – Previous Month Reading]

Step 4 If a connection type is domestic goto step 6

Step 5 Else goto step 7

Step 6 Initialize i with 1 and sum as 0

> **Step 6.1** If check i is less than or equal to 100 then compute sum = sum + 1 and gotostep 6.5
>
> **Step 6.2** Else if check i is greater than 100 and less than 200 then compute sum = sum+ 2.5 and gotostep 6.5
>
> **Step 6.3** Else if check i is greater than 200 and less than 500 then compute sum = sum + 4 and goto step6.5
>
> **Step 6.4** Else compute sum = sum + 6 and goto step 6.5
>
> **Step 6.5** If i is equal to number of units consumed goto step 8 else return to same step

Step 7 Initialize i with 1 and sum as 0

> **Step 7.1** If check i is less than or equal to 100 then compute sum = sum + 2 and goto step 7.5
>
> **Step 7.2** Else if check i is greater than 100 and less than 200 then compute sum = sum + 4.5 and gotostep 7.5
>
> **Step 7.3** Else if check i is greater than 200 and less than 500 then compute sum = sum+ 6 and goto step7.5
>
> **Step 7.4** Else compute sum = sum + 7 and goto step 7.5
>
> **Step 7.5** If i is equal to number of units consumed goto step 8 else return to same step

Step 8 Store the sum

Step 9 Display Bill Details [Name, Consumer Number, No of units consumed]

Step 10 Check the connection type

> **Step 10.1** If connection type is domestic display domestic tariff slot and goto step 11
>
> **Step 10.2** Else display commercial tariff slot and goto step 11

Step 11 Display amount payable using stored sum

Step 12 Stop the Process

Program:

```java
import java.io.*;
public class ElectricityBill
{
public static void main() throws IOException
{
BufferedReader br=new BufferedReader (new InputStreamReader (System.in));
System.out.println("Enter Your Name : ");
String nm=br.readLine();
System.out.println("Enter the units consumed : ");
int units=Integer.parseInt(br.readLine());
double charges=0,fbill=0;
System.out.println("Please enter your choice:");
System.out.println("Domestic for 1 && Consumer for 2 ");
int choice=in.nextInt();
switch (choice)
{
case 1:
if(units<=100)
charges=units*1;
else if(units<=101&&units>=200)
charges=(100*1)+(units-100)*2.50;
else if(units<=201&&units>=500)
charges=(100*1)+(100*2.50)+(units-500)*4;
else
charges=(100*1)+(100*2.50)+(500*4)+(units-500)*6;
fbill=charges+250;
System.out.println("\f");
break;
case 2:
if(units<=100)
charges=units*2;
else if(units<=101&&units>=200)
charges=(100*2)+(units-100)*4.50;
else if(units<=201&&units>=500)
charges=(100*2)+(100*4.50)+(units-500)*6;
else
charges=(100*2)+(100*4.50)+(500*6)+(units-500)*7;
fbill=charges+250;
System.out.println("\f");
break;
}
//printing the bill
System.out.println("Your Electricity Bill");
System.out.println("********************");
System.out.println();
System.out.println("Costumers Name : "+nm);
System.out.println("Units Consumed : "+units);
System.out.println("Charges : "+charges);
System.out.println("Rent Charges : 250 Rs.");
System.out.println("Total Bill : "+fbill);
}
}
```

Output:

```
C:\Users\admin\Desktop>java ElectricityBill
Enter Your Name :
Senthil
Enter the units consumed :
100
Please enter your choice:
Domestic for 1 && Consumer for 2
2
9
Your Electricity Bill
xxxxxxxxxxxxxxxxxxxxxx

Costumers Name : Senthil
Units Consumed : 100
Charges : 200.0
Rent Charges : 250 Rs.
Total Bill : 450.0

C:\Users\admin\Desktop>_
```

Result:

  The Java console application for electricity bill generator was developed and tested successfully.

| **Ex.No: 2** | **Fahrenheit to Celsius degree** |
| **Date:** | |

Aim:

Implement a program to convert temperature from Fahrenheit to Celsius degree by using the formula given below and display the converted value.

C = ((F-32)/9)*5 where, C represents temperature in Celsius and F represents temperature in Fahrenheit.

Algorithm:

STEP 1: Initialize the code

STEP 2: declare the variables and their respective values.

STEP 3: Run the code in jdk , after execution the output is displayed.

STEP 4: End of execution

Program :

```
import java.io.*;
class Simple
{
public static void main(String[] args)
{
double F=64;
double C = ((F-32)/9)*5 ;
System.out.println(C);
}
}
```

Output :

```
C:\Program Files\Java\jdk1.8.0_202\bin>javac Simple.jav

C:\Program Files\Java\jdk1.8.0_202\bin>java Simple
17.77777777777778
```

Result :

Thus a java program has been written for conversion of celsius to Fahrenheit.

Ex. No: 3 **Date:**	**Unit Converters Application using packages**

Aim:

Develop a java application to implement currency converter (Dollar to INR, EURO to INR, Yen to INR and vice versa), distance converter (meter to KM, miles to KM and vice versa), time converter (hours to minutes, seconds and vice versa) using packages.

Algorithm:

Step 1 Start the process

Step 2 Prompt the user with converter choice 1. Currency 2.Distance 3. Time 4. Exit thechoice.

Step 3 If user selects a Currency Converter then proceed to step 4

Step 4 Proceed with prompting Currency Converter Choices

    1. DOLLER to INR 2. EURO to INR 3. YEN to INR

    4. INR to DOLLER 5. INR to EURO 6. INR to YEN

    7. Exit and get the user choice.

        Step 4.1 If option 1 selected get in DOLLER and display DOLLER * 66.89 as INR

        Step 4.2 If option 2 selected get in EURO and display EURO * 80 as INR

        Step 4.3 If option 3 selected get in YEN and display YEN * 0.61 as INR

        Step 4.4 If option 4 selected get in INR and display INR / 66.89 as DOLLER

        Step 4.5 If option 5 selected get in INR and display INR / 80 as EURO

        Step 4.6 If option 6 selected get in INR and display INR / 0.61 as YEN

        Step 4.7 If option 7 selected exit from currency converter choice goto step 2

Step 5 If user selects a Distance Converter then proceed to step 6

Step 6 Proceed with prompting Distance Converter Choices

1. METER to KILOMETER 2. MILES to KILOMETER

3. KILOMETER to METER 4. KILOMETER to MILES

5. Exit and get the user choice.

        Step 6.1 If option 1 selected get in METER and display METER / 1000 as KILOMETER

        Step 6.2 If option 2 selected get in MILES and display MILES * 1.60934 as KILOMETER

        Step 6.3 If option 3 selected get in KILOMETER and display KILOMETER * 1000 as METER

        Step 6.4 If option 4 selected get in KILOMETER and display KILOMETER / 1.60934 as MILES

        Step 6.5 If option 5 selected exit from distance converter choice goto step 2

Step 7 If user selects a Time Converter then proceed to step 8 currency

Step 8 Proceed with prompting Time Converter Choices

1. HOURS to MINUTES 2. HOURS to SECONDS

3. MINUTES to HOURS 4. SECONDS to HOURS

5. Exit and get the user choice.

        Step 8.1 If option 1 selected get in HOURS and display HOURS * 60 as MINUTES

        Step 8.2 If option 2 selected get in HOURS and display HOURS * 3600 as SECONDS

        Step 8.3 If option 3 selected get in MINUTES and display MINUTES / 60 as HOURS

        Step 8.4 If option 4 selected get in SECONDS and display SECONDS / 3600 as HOURS.

        Step 8.5 If option 5 selected exit from tim converter choice and goto step 2

Step 9 If user selects exit then display Thank You !!! and exit from the system

Step 10 Stop the process.

Program: Currency Converter

```java
import java.io.*;
import java.util.Scanner;
public class Currency {
public static double covertEUROtoINR(double EURO) {
return EURO * 80;
}
public static double convertDOLLARtoINR(double DOLLAR) {
return DOLLAR * 66.89;
}
public static double convertYENtoINR(double YEN) {
return YEN * 0.61;
}
public static double covertINRtoEURO

(double INR) {
return INR * 0.013;
}
public static double convertINRtoDOLLAR(double DOLLAR) {
return DOLLAR * 0.015;
}
public static double convertINRtoYEN(double YEN) {
return YEN * 1.63;
}
public static void userChoice(){
Scanner input = new Scanner(System.in);
int currency_choice = 0;
double money = 0;
while(currency_choice != 7){
System.out.println("\nCurrency Converter");
System.out.println(" ------------------");
System.out.println("1. DOLLOR to INR\n2.EURO to INR\n3. YEN to INR\n"+ "4. INR to DOLLOR\n5. INR to
EURO\n6.INR to YEN\n"+ "7.Exit\n\nEnter Your Choice");
currency_choice = input.nextInt();
switch(currency_choice){
case 1:
System.out.println("Enter in DOLLER");
money = input.nextDouble();
System.out.println(money+"DOLLER is equal to"+Currency.convertDOLLARtoINR(money)+"INR");
break;
case 2:
System.out.println("Enter in EURO");
money = input.nextDouble();
System.out.println(money+" EURO is equal to"+Currency.covertEUROtoINR(money)+"INR");
break;
case 3:
System.out.println("Enter in YEN");
money = input.nextDouble();
System.out.println(money+" YEN is equal to"+Currency.convertYENtoINR(money)+"INR");
break;
case 4:
System.out.println("Enter in INR");
money = input.nextDouble();
System.out.println(money+" INR is equal to"+Currency.convertINRtoDOLLAR(money)+"DOLLORS");
break;
case 5:
System.out.println("Enter in INR");
money = input.nextDouble();
System.out.println(money+" INR is equal to"+Currency.covertINRtoEURO(money)+" EURO");
break;
case 6:
System.out.println("Enter in INR");
money = input.nextDouble();
System.out.println(money+" INR is equal to "+Currency.convertINRtoYEN(money)+" YEN");
break;
case 7:
break;
default:
System.out.println("Please choose valid option");
```

```
break;
}
}
}
}
```

Program: Distance Converter

```java
import java.io.*;
import java.util.Scanner;
public class Distance {
public static double convertMeterToKiloMeter(double meter) {
return meter / 1000;
}
public static double convertMilesToKiloMeter(double miles) {
return miles * 1.60934;
}
public static double convertKiloMetertoMeter(double kilometer) {
return kilometer * 1000;
}
public static double convertKiloMeterToMiles(double kilometer) {
return kilometer / 1.60934;
}
public static void userChoice(){
Scanner input = new Scanner(System.in);
int distance_choice = 0;
double distance = 0;
while(distance_choice != 5){
System.out.println("\nDistance Converter");
System.out.println(" ---------------- ");
System.out.println("1. METER to KILOMETER\n2. MILES to KILOMETER\n"+ "3. KILOMETER to
METER\n4. KILOMETER to MILES\n"+ "5.Exit\n\nEnter Your Choice");
distance_choice = input.nextInt();
switch(distance_choice){
case 1:
System.out.println("Enter in METERS");
distance = input.nextDouble();
System.out.println(distance+" METERS is equal
to"+Distance.convertMeterToKiloMeter(distance)+"KILOMETER");
break;
case 2:
System.out.println("Enter in MILES");
distance = input.nextDouble();
System.out.println(distance+" MILES is equal
to"+Distance.convertMilesToKiloMeter(distance)+"KILOMETER");
break;
case 3:
System.out.println("Enter in KILOMETER");
distance = input.nextDouble();
System.out.println(distance+" KILOMETER is equal
to"+Distance.convertKiloMetertoMeter(distance)+"METER");
break;
case 4:
System.out.println("Enter in KILOMETER");
distance = input.nextDouble();
System.out.println(distance+" KILOMETER is equal
to"+Distance.convertKiloMeterToMiles(distance)+"MILES");
break;
case 5:
break;
default:
System.out.println("Please choose valid option");
break;
}
```

```
        }
    }
}
```

Program: Time Converter

```
import java.io.*;
import java.util.Scanner;
public class Time {
public static double convertHoursToMinutes(double hours) {
return hours * 60;
}
public static double convertHoursToSeconds(double hours) {
return hours * 60 * 60;
}
public static double convertMinutesToHours(double minutes) {
return minutes / 60;
}
public static double convertSecondsToHours(double seconds) {
return seconds / 60 / 60;
}
public static void userChoice(){
Scanner input = new Scanner(System.in);
int time_choice = 0;
double time = 0;
while(time_choice != 5){
System.out.println("\nTime Converter");
System.out.println(" ---------------- ");
System.out.println("1. HOURS to MINUTES\n2. HOURS to SECONDS\n"+ "3. MINUTES to
HOURSn4.SECONDS to HOURS\n"+ "5.Exit\n\nEnter Your Choice");
time_choice = input.nextInt();
switch(time_choice){
case 1:
System.out.println("Enter in HOURS");
time = input.nextDouble();
System.out.println(time+" HOURS is equal to"+Time.convertHoursToMinutes(time)+"MINUTES");
break;
case 2:
System.out.println("Enter in HOURS");
time = input.nextDouble();
System.out.println(time+" HOURS is equal to"+Time.convertHoursToSeconds(time)+"SECONDS");
break;
case 3:
System.out.println("Enter in MINUTES");
time = input.nextDouble();
System.out.println(time+" MINUTES is equal to"+Time.convertMinutesToHours(time)+"HOURS");
break;
case 4:
System.out.println("Enter in SECONDS");
time = input.nextDouble();
System.out.println(time+" SECONDS is equal to"+Time.convertSecondsToHours(time)+"HOURS");
break;
case 5:
break;
default:
System.out.println("Please choose valid option");
break;
}
}
}
}
```

Program: Main

```java
import com.Converters.Currency;
import com.Converters.Distance;
import com.Converters.Time;
import java.io.*;
import java.util.Scanner;
public class Main {
public static void main(String[] args) {
Scanner input = new Scanner(System.in);
int choice = 0;
while(choice != 4){
System.out.println("Converters");
System.out.println("**********");
System.out.println("1. Currency\n2. Distance\n3. Time\n4. Exit\n\nEnter Your Choice");
choice = input.nextInt();
switch(choice){
case 1:
Currency.userchoice();
break;
case 2:
Distance.userchoice();
break;
case 3:
Time.userchoice();
break;
case 4:
break;
default:
System.out.println("Please choose valid option");
break;
}
}
System.out.println("Thank You !!!!");
}
}
```

Output:

```
Main (10) [Java Application] /usr/lib/jvm/java-8-openjdk-amd64/bin/java (31-May-2018, 11:56:17 AM)
Converters
**********
1. Currentcy
2. Distance
3. Time
4. Exit

Enter Your Choice
1

Currency Converter
------------------
1. DOLLOR to INR
2. EURO to INR
3. YEN to INR
4. INR to DOLLOR
5. INR to EURO
6. INR to YEN
7.Exit

Enter Your Choice
1
Enter in DOLLER
10
10.0 DOLLER is equal to 668.9 INR
```

```
Console ☒
Main (10) [Java Application] /usr/lib/jvm/java-8-openjdk-amd64/bin/java (31-May-2018, 11:59:41 AM)
Converters
**********
1. Currentcy
2. Distance
3. Time
4. Exit

Enter Your Choice
3

Time Converter
----------------
1. HOURS to MINUTES
2. HOURS to SECONDS
3. MINUTES to HOURS
4. SECONDS to HOURS
5.Exit

Enter Your Choice
4
Enter in SECONDS
77378728
7.7378728E7 SECONDS is equal to 21494.09111111111 HOURS
```

```
Console ☒
Main (10) [Java Application] /usr/lib/jvm/java-8-openjdk-amd64/bin/java (31-May-2018, 11:58:07 AM)
Converters
**********
1. Currentcy
2. Distance
3. Time
4. Exit

Enter Your Choice
2

Distance Converter
------------------
1. METER to KILOMETER
2. MILES to KILOMETER
3. KILOMETER to METER
4. KILOMETER to MILES
5.Exit

Enter Your Choice
1
Enter in METERS
2789
2789.0 METERS is equal to 2.789 KILOMETER
```

Result

The java console application for converters [Currency, Distance, and Time] was developed and tested successfully.

| **Ex.No: 4** | **Employee Payroll System** |
| **Date:** | |

Aim:

Develop a java application with Employee class with Emp_name, Emp_id, Address, Mail_id, Mobile_no as members. Inherit the classes, Programmer, Assistant Professor, Associate Professor and Professor from employee class. Add Basic Pay (BP) as the member of all the inherited classes with 97% of BP as DA, 10 % of BP as HRA, 12% of BP as PF, 0.1% of BP for staff club fund. Generate pay slips for the employees with their gross and net salary.

Algorithm:

Step 1 Start the process

Step 2 Prompt the user with converter choice 1. Programmer 2. Assistant Professor 3. Associate Professor 4. Professor 5. Exit and get the choice.

Step 3 If user selects a Programmer then proceed to step 4

Step 4 Get the user details [Name, ID, Address, Mail ID and Mobile Number] and goto step 5

Step 5 Proceed to Programmers Payment Processing

    Step 5.1 Get the basic pay of Programmer

    Step 5.2 If user enters -1 assume basic pay as 30000 and goto step 15

    Step 5.3 Else directly go to step 15

Step 6 If user selects Assistant Professor step 7

Step 7 Get the user details [Name, ID, Address, Mail ID and Mobile Number] and goto step 8

Step 8 Proceed to Assistant Professor Payment Processing

    Step 8.1 Get the basic pay of Assistant Professor

    Step 8.2 If user enters -1 assume basic pay as 25000 and goto step 15

    Step 8.3 Else directly go to step 15

Step 9 If user selects Associate Professor step 10

Step 10 Get the user details [Name, ID, Address, Mail ID and Mobile Number] and goto step 11

Step 11 Proceed to Associate Professor Payment Processing

    Step 11.1 Get the basic pay of Associate Professor

    Step 11.2 If user enters -1 assume basic pay as 40000 and goto step 15

    Step 11.3 Else directly go to step 15

Step 12 If user selects Professor step 13

Step 13 Get the user details [Name, ID, Address, Mail ID and Mobile Number] and goto step 14

Step 14 Proceed to Professor Payment Processing

    Step 14.1 Get the basic pay of Professor

    Step 14.2 If user enters -1 assume basic pay as 70000 and goto step 15

    Step 14.3 Else directly go to step 15

Step 15 Compute Per_Day_Pay = original_basic_pay / no_of_days_in_the_current_month

Step 16 Get the number of days worked from user that include Cl, WH, FH and exclude the LOP

Step 17 Check no_days_worked <= no_of_days_in_the_current_month. Else display "Error Message" and goto step 18

Step 18 Compute Current_Basic_Pay = Per_Day_Pay * no_days_worked

Step 19 Compute Following and Store

    DA = (Current_Basic_Pay/100) * 97

    HRA = (Current_Basic_Pay/100) * 12

    PF = (Current_Basic_Pay/100) * 0.1

    GROSS_PAY = Current_Basic_Pay + DA + HRA + PF

    NET_PAY = GROSS_PAY - PF

    Step 17 Display Payment Details [Name, ID, Address, Mail ID, Mobile Number, BASIC PAY,DA, HRA,PF,GROSS_PAY, NET_PAY].

Step 18 Stop Processing
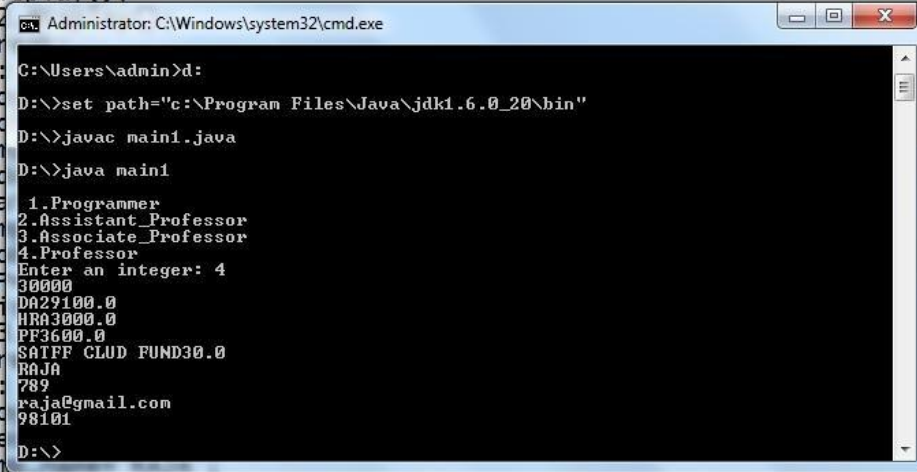
Program:
```
import java.util.Scanner;
class Employee
{
String Emp_name;
int Emp_id;
String Address;
String Mail_id;
int Mobile_no;
void display(){
System.out.println(Emp_name);
//Syetem.out.println(Address);
System.out.println(Emp_id);
System.out.println(Mail_id);
System.out.println(Mobile_no);
}}
class Programmer extends Employee{
 int BP;
/*int DA= (int) (0.97*BP);
HRA=(int) (0.10*BP);
PF=(int) (0.12*BP);  */
void display(){
   System.out.println(BP);
 System.out.println("DA"+0.97*BP);
 System.out.println("HRA"+0.10*BP);
   System.out.println("PF"+0.12*BP);
 system.out.println("satff clud fund"+0.001*bp);
}}
class assistant_professor extends employee{
 int bp;
void display(){
   system.out.println(bp);
 system.out.println("da"+0.97*bp);
 system.out.println("hra"+0.10*bp);
   system.out.println("pf"+0.12*bp);
 system.out.println("satff clud fund"+0.001*bp);
}}
class associate_professor extends employee{
 int bp;
void display(){
   system.out.println(bp);
 system.out.println("da"+0.97*bp);
 system.out.println("hra"+0.10*bp);
   system.out.println("pf"+0.12*bp);
 system.out.println("satff clud fund"+0.001*bp);
}}
class professor extends employee{
 int bp;
void display(){
   system.out.println(bp);
 system.out.println("da"+0.97*bp);
 system.out.println("hra"+0.10*bp);
   system.out.println("pf"+0.12*bp);
 system.out.println("satff clud fund"+0.001*bp);
}}
class main1{
public static void main(string args[]){
   system.out.println("\n 1.programmer\n2.assistant_professor\n3.associate_professor\n4.professor");
scanner input=new scanner(system.in);
system.out.print("enter an integer: ");
int ch=input.nextint();
 switch (ch) {
      case 1:
```

```java
        employee e1=new employee();
      programmer p1=new programmer();
      e1.emp_name="abc";
      e1.address="y-city";
      e1.mail_id="praw@gmail.com";
      e1.emp_id=567;
      e1.mobile_no=2345678;
      p1.bp=15000;
      p1.display();
        e1.display();
        break;
    case 2:
      employee e2=new employee();
      assistant_professor p2=new assistant_professor();
      e2.emp_name="def";
      e2.address="a-city";
      e2.mail_id="rajan@gmail.com";
      e2.emp_id=123;
      e2.mobile_no=987321;
      p2.bp=30000;
      p2.display();
        e2.display();
        break;
    case 3:
      employee e3=new employee();
      associate_professor p3=new associate_professor();
      e3.emp_name="ghf";
      e3.address="b-city";
      e3.mail_id="main@gmail.com";
      e3.emp_id=456;
      e3.mobile_no=98710;
      p3.bp=30000;
      p3.display();
        e3.display();
        break;
    case 4:
      employee e4=new employee();
      professor p4=new professor();
      e4.emp_name="kannan";
      e4.address="trichy";
      e4.mail_id="kanna@gmail.com";
      e4.emp_id=789;
      e4.mobile_no=9810;
      p4.bp=30000;
      p4.display();
        e4.display();
        break;
    case 5:
     //exit(1);
     default:
     system.out.println("enter correct choice");

} }}
```

Output:


```
C:\Users\admin>d:

D:\>set path="c:\Program Files\Java\jdk1.6.0_20\bin"

D:\>javac main1.java

D:\>java main1

 1.Programmer
2.Assistant_Professor
3.Associate_Professor
4.Professor
Enter an integer: 4
30000
DA29100.0
HRA3000.0
PF3600.0
SATFF CLUD FUND30.0
RAJA
789
raja@gmail.com
98101

D:\>
```

Result:

The java console application for employee payroll system was developed and tested successfully.

| **Ex.No: 5** | **Java Program for Class and Objects** |
| **Date:** | |

Aim:
Create a new class Order in the Java project SwiftFood with the instance variables and methods mentioned below.
- instance variable:
  - ➢ orderid-int
  - ➢ orderedfood-String
  - ➢ totalprice-double
  - ➢ status-String.

Method - calculateTotalPrice(int unitPrice):double

Method Description
- ➢ Calculate the total price by applying a service charge of 5% on the food item ordered and store it in the instance variable totalPrice.
- ➢ Return the calculated total price.
- ➢ Create an object of the Order class, initialize the instance variables, invoke the calculateTotalPrice() method and display the values of the instance variables in the main() method of the Tester class.

Algorithm:
Step 1: Initialize execution
Step 2: Enter the value of n & the condition will be checked
Step 3 : After checking the answer will return either yes or no
Step 4 : End of execution

Program :

```java
package exercise;
import java.io.*;
class Order {
private int orderid;
private String orderedfood;
private double totalprice;
private String status;
public Order(int orderid, String orderedfood) {
this.orderid = orderid;
this.orderedfood = orderedfood;
}
public double calculateTotalPrice(int unitPrice) {
double totalPrice = unitPrice * 1.05; // Applying a service charge of 5%
this.totalprice = totalPrice;
return totalPrice;
}
// Getters and setters for the instance variables
public int getOrderid() {
return orderid;
}
public void setOrderid(int orderid) {
this.orderid = orderid;
}
public String getOrderedfood() {
return orderedfood;
}
public void setOrderedfood(String orderedfood) {
this.orderedfood = orderedfood;
}
public double getTotalprice() {
return totalprice;
}
public void setTotalprice(double totalprice) {
this.totalprice = totalprice;
}
public String getStatus() {
return status;
}
public void setStatus(String status) {
this.status = status;
} }
```

```java
public class Farm {
public static void main(String[] args) {
Order order = new Order(1, "Pizza");
double totalPrice = order.calculateTotalPrice(10);
order.setStatus("Pending");
System.out.println("Order ID: " + order.getOrderid());
System.out.println("Ordered Food: " + order.getOrderedfood());
System.out.println("Total Price: " + order.getTotalprice());
System.out.println("Status: " + order.getStatus());
}
}
```

<u>Output:</u>

Order ID: 1

Ordered Food: Pizza

Total Prize: 100

Status: Pending

<u>Result:</u>

    Thus a java code has been written and the output has been displayed.

Ex.No: 6	Java interface for ADT Stack
Date:	

Aim:

Design a Java interface for ADT Stack. Implement this interface using array. Provide necessary exception handling in both the implementations.

Algorithm:
Step 1 Start the process.
Step 2 Get the Stack's maximum limit from user.
Step 3 Create an array with the limit and initialize all the elements as -1 and initialize current_position with 0
Step 4 Prompt the user with choice for stack operations
   1. PUSH 2.POP 3.PEEK 4.DISPLAY 5.EXIT
Step 5 Get the choice from user and goto step 6
Step 6 If user selects to PUSH
  Step 6.1 Get the number from user to push.
  Step 6.2 Try to assign the value to array assuming that array index is current position.
  Step 6.3 If exception rises display message "Stack is full try to pop someting" and goto step 9.
  Step 6.4 Else increment current position with the value 1 and display element pushed index and goto
step 9.
Step 7 If user selects to POP
  Step 7.1 Try to set previous position of current position to -1
  Step 7.2 If exception rises display message "Stack is empty try to push something" and goto step 9.
  Step 7.3 Else decrement current position with the value 1 and display popped element and goto step 9.
Step 8 If user selects to PEEK display element display current_position -1 element in the array
Step 9 If user selects to DISPLAY
  Step 9.1 Display array reversely and elements which is not equal to -1 and goto step 4
Step 10 Exit from the process.
Step 11 Stop the process.

Program:

```java
import java.io.*;
interface Mystack
{
    public void pop();
    public void push();
    public void display();
}
class Stack_array implements Mystack
{
    final static int n=5;
    int stack[]=new int[n];
    int top=-1;
    public void push()
    {
        try
        {
            BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
            if(top==(n-1))
            {
                System.out.println(" Stack Overflow");
                return;
            }
            else
            {
                System.out.println("Enter the element");
                int ele=Integer.parseInt(br.readLine());
                stack[++top]=ele;
            }
        }
        catch(IOException e)
        {
            System.out.println("e");
        }   }
    public void pop()
    {
        if(top<0)
        {
            System.out.println("Stack underflow");
            return;
        }
        else
        {
            int popper=stack[top];
            top--;
            System.out.println("Popped element:" +popper);
        }   }
     public void display()
    {
        if(top<0)
        {
            System.out.println("Stack is empty");
            return;
        }
        else
        {
            String str=" ";
            for(int i=0; i<=top; i++)
                str=str+" "+stack[i]+" <--";
            System.out.println("Elements are:"+str);
        }  }}
class Link
```

```java
    {
        public  int  data;
        public Link nextLink;
        public Link(int d)
        {
            data= d;
            nextLink=null;
        }
        public void printLink()
        {
            System.out.print(" --> "+data);
        }}
class Stack_List  implements Mystack
{
        private Link first;
        public Stack_List()
        {
            first = null;
        }
        public boolean isEmpty()
        {
            return first == null;
        }
        public void push()
        {
            try
            {
                BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
                System.out.println("Enter the element");
                int ele=Integer.parseInt(br.readLine());
                Link link = new Link(ele);
                link.nextLink = first;
                first = link;
            }
            catch(IOException e)
            {
                System.err.println(e);
            }   }
        public Link delete()
        {
            Link temp = first;
            try
            {
                first = first.nextLink;
            }
            catch(NullPointerException e)
            {
                throw e;
            }
            return temp;
        }
        public void pop()
        {
            try
            {
                Link deletedLink = delete();
                System.out.println("Popped: "+deletedLink.data);
            }
            catch(NullPointerException e)
            {
                throw e;
            }   }
        public void display()
```

```java
      {
         if(first==null)
            System.out.println("Stack is empty");
         else
         {
            Link currentLink = first;
            System.out.print("Elements are: ");
            while(currentLink != null)
            {
               currentLink.printLink();
               currentLink = currentLink.nextLink;
            }
            System.out.println("");
      }    }
}
class StackADT
{
   public static void main(String arg[])throws IOException
   {
      BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
      System.out.println("Implementation of Stack using Array");
      Stack_array stk=new Stack_array();
      int ch=0;
      do
      {
         System.out.println("1.Push 2.Pop 3.Display 4.Exit 5.Use Linked List");
         System.out.println("Enter your choice:");
         ch=Integer.parseInt(br.readLine());
         switch(ch)
         {
         case 1:
            stk.push();
            break;
         case 2:
            stk.pop();
            break;
         case 3:
            stk.display();
            break;
         case 4:
            System.exit(0);
      }       }
      while(ch<5);
      System.out.println("Implementation of Stack using Linked List");
      Stack_List stk1=new Stack_List();
      ch=0;
      do
      {
         System.out.println("1.Push 2.Pop 3.Display 4.Exit");
         System.out.println("Enter your choice:");
         ch=Integer.parseInt(br.readLine());
         switch(ch)
         {
         case 1:
            stk1.push();
            break;
         case 2:
            try
            {
               stk1.pop();
            }
            catch(NullPointerException e)
            {
```
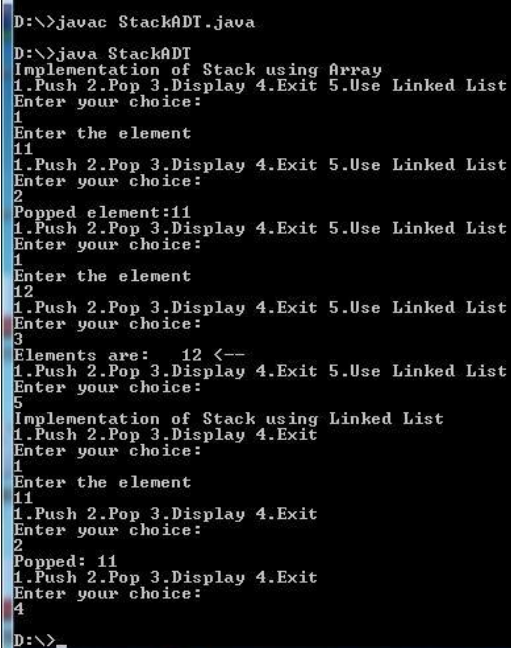
```
            System.out.println("Stack underflown");
          }
          break;
      case 3:
          stk1.display();
          break;
      default:
          System.exit(0);
        }
    }
  while(ch<5);
}}
```

Output:

Ex.No: 7 **Date:**	**Java Application for String Operations using ArrayList**

Aim:

Write a program to perform string operations using ArrayList. Write functions for the following

    a.   Append - add at end

    b.   Insert – add at particular index

    c.   Search

    d.   List all string starts with given letter

Algorithm:

Step 1 Start the Process

Step 2 Prompt user with List Operation Choices

       1. Append 2. Insert at particular Index 3. Search a string in a list

       4. Display all the strings in a list 5. Display all the strings that begins with a character

       6. Exit

Step 3 If user selects option 1 then get the string from user and add at last position of the list.

Step 4 If user selects option 2

       Step 4.1 Get the string and position from the user

       Step 4.2 Check the position exceeds from list size

       Step 4.3 If exceeds then prompt an error "Sorry Position is not in the list limit" and goto step 2.

       Step 4.4 Else create temporary list

       Step 4.5 Add all the elements of the list upto current position one by one.

       Step 4.6 Add current element to the list

       Step 4.7 Add remaining elements one by one.

       Step 4.8 Assign temporary list as source list and goto step 2

Step 5 If user selects option 3 then get the string to be searched from the user.

       Step 5.1 Check every string one by one for string match

       Step 5.2 If match found and display the string and position and goto step 2

       Step 5.3 Else prompt an error message "String not found in the list" and goto step 2

Step 6 If user selects option 4 the display all the strings one by one. After that goto step 2.

Step 7 If user selects option 5 then proceed following

       Step 7.1 Get the character from user

       Step 7.2 Get all the string one by one and match first letter with character entered by the user

       Step 7.3 If character matches display the string

       Step 7.4 Else move to next string.

       Step 7.5 At the end goto step 2

Step 8 If user selects option 6 exit from process and stop processing

Program:

```java
import java.io.*;
import java.util.ArrayList;
public class MyBasicArrayList
{
   public static void main(String args[])
{
        String prefix="P";
    ArrayList<String> al = new ArrayList<String>();
    ArrayList<String> result = new ArrayList<String>();
    //add elements to the ArrayList
    al.add("JAVA");
    al.add("C++");
    al.add("PERL");
    al.add("PHP");
    System.out.println(al);
    //get elements by index
    System.out.println("Element at index 1: "+al.get(1));
    System.out.println("Does list contains JAVA? "+al.contains("JAVA"));
    //add elements at a specific index
    al.add(2,"PLAY");
    System.out.println(al);
    System.out.println("Is arraylist empty? "+al.isEmpty());
    System.out.println("Index of PERL is "+al.indexOf("PERL"));
    System.out.println("Size of the arraylist is: "+al.size());
        for(String s: al) {
     if(s.startsWith(prefix))
       {
      result.add(s);
    }
    }
    System.out.println(result);
   }
}
```

Output:


```
D:\>javac arraylist1.java

D:\>java arraylist1
[JAVA, C++, PERL, PHP]
Element at index 1: C++
Does list contains JAVA? true
[JAVA, C++, PLAY, PERL, PHP]
Is arraylist empty? false
Index of PERL is 3
Size of the arraylist is: 5
[PLAY, PERL, PHP]

D:\>
```

Result:

The java console application for string list operations was developed and tested successfully.

| **Ex.No: 8** | **Java Application using Getter and Setter Methods** |
| **Date:** | |

Aim:

Create a class Student that has a data member rollno and getter and setter methods to get and set rollno. Derive a class Test from Student that has data members marks1, marks2 and getter and setter methods to get and set the marks. Create a class Result derived from class Test that has a datamember total and a method display() to display the members of all the classes. From main() method, display the members of all the classes.

Algorithm:

STEP 1: Initialize the code

STEP 2: Enter the credentials to be displayed

STEP 3: Run the code in jdk and output is displayed
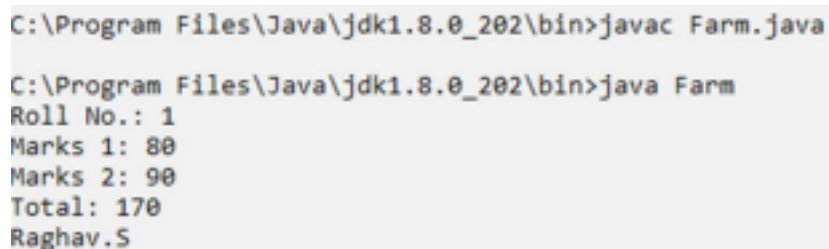
STEP 4: End of execution

PROGRAM:

```
class Student {

private int rollno;

public void setRollNo(int rollno) {

this.rollno = rollno;

}

public int getRollNo() {

return rollno;

}}

class Test extends Student {

private int marks1;

private int marks2;

public void setMarks1(int marks1) {

this.marks1 = marks1;
}

public int getMarks1() {

return marks1;

}

public void setMarks2(int marks2) {

this.marks2 = marks2;

}

public int getMarks2() {

return marks2;

}

}
```

```java
class Result extends Test {

 private int total;

 public void calculateTotal() {

 total = getMarks1() + getMarks2();   }

 public void display() {

 System.out.println("Roll No.: " + getRollNo());   System.out.println("Marks 1:

"  +  getMarks1());      System.out.println("Marks  2:  "  +  getMarks2());

System.out.println("Total: " + total);

 System.out.println("K.Sathaiah");

 }

 }
public class Main {

 public static void main(String[] args) {

 Result result = new Result();

 result.setRollNo(1);

 result.setMarks1(80);

 result.setMarks2(90);

 result.calculateTotal();

 result.display();

 }

 }
```

OUTPUT:

```
C:\Program Files\Java\jdk1.8.0_202\bin>javac Farm.java

C:\Program Files\Java\jdk1.8.0_202\bin>java Farm
Roll No.: 1
Marks 1: 80
Marks 2: 90
Total: 170
Raghav.S
```

RESULT:

Thus we have executed the program and the expected output is displayed.

| **Ex.No: 9** | **Java Application to Find the Area of different Shapes** |
| **Date:** | |

Aim:

Write a Java Program to create an abstract class named Shape that contains two integers and an empty method named print Area(). Provide three classes named Rectangle, Triangle and Circle such that each one of theclasses extends the class Shape. Each one of the classes contains only the method print Area () that prints the areaof the given shape.

Algorithm:

Step 1 Start the Process

Step 2 Prompt user with List Operation Choices

1. Rectangle 2. Triangle 3. Circle 4, Exit

Get the choice from user.

Step 3 If user selects Rectangle

Step 3.1 Get the Length and Breath from the user

Step 3.2 Compute Area = Length * Breath

Step 3.3 Display Area

Step 4 If user selects Triangle

Step 4.1 Get the Length and Height from the user

Step 4.2 Compute Area = Length * Height * 0.5

Step 4.3 Display Area

Step 5 If user selects Circle

Step 5.1 Get the Radius of the Circle

Step 5.2 Compute Area = 3.14 * Radius * Radius

Step 5.3 Display Area

Step 6 If user selects exit end the process

Step 7 Stop the Process

Program:

```
abstract class shape
{
int a=3,b=4;
 abstract public void print_area();
}
class rectangle extends shape
{
 public int area_rect;
 public void print_area()
 {
  area_rect=a*b;
System.out.println("The area of rectangle is:"+area_rect);
 }
}
class triangle extends shape
{
 int area_tri;
 public void print_area()
 {
  area_tri=(int) (0.5*a*b);
System.out.println("The area of triangle is:"+area_tri);
 }
}
class circle extends shape
{
 int area_circle;
 public void print_area()
 {
  area_circle=(int) (3.14*a*a);
System.out.println("The area of circle is:"+area_circle);
 }
}
public class JavaApplication3
{
public static void main(String args[])
{
     rectangle r=new rectangle();
     r.print_area();
     triangle t=new triangle();
     t.print_area();
     circle r1=new circle();
     r1.print_area();
}
}
```

```
D:\>javac JavaApplication3.java

D:\>java JavaApplication3
The area of rectangle is:12
The area of triangle is:6
The area of circle is:28

D:\>
```

Result:

　　　　The Java console application to find the area of different shapes using abstract class concept in java was developed and tested successfully.

Ex.No: 10	**Creating User Defined Exceptions**
Date:	

Aim:

Write a Java program to implement user defined exception handling.

Algorithm:

Step 1 Start the Process

Step 2 Prompt user with input options.

Step 3 If user gives correct options

      Step 3.1 Execute the statements.

Step 4 If user gives wrong options

      Step 4.1 Execute the user defined exceptions created.

Step 5 Stop the Process

Program:

```java
import java.util.*;
import java.io.*;
class MyException extends Exception{
   String str1;
  /* Constructor of custom exception class
   * here I am copying the message that we are passing while
   * throwing the exception to a string and then displaying
   * that string along with the message.
   */
  MyException(String str2) {
          str1=str2;
  }
  public String toString(){
          return ("MyException Occurred: "+str1) ;
  }
}

class Example1{
  public static void main(String args[]){
          try{
                  System.out.println("Starting of try block");
                  // I'm throwing the custom exception using throw
                  throw new MyException("This is My error Message");
          }
          catch(MyException exp){
                  System.out.println("Catch Block") ;
                  System.out.println(exp) ;
          }
  }
}
```

Output:

```
D:\>set path="c:\Program Files\Java\jdk1.6.0_20\bin"

D:\>javac Example1.java

D:\>java Example1
Starting of try block
Catch Block
MyException Occurred: This is My error Message

D:\>_
```

Result:

        The java console application that uses user defined exception handling techniques was developed and tested successfully.

```
D:\>set path="c:\Program Files\Java\jdk1.6.0_20\bin"

D:\>javac Example1.java
```

| **Ex.No: 11** | **Java Application for File Handling** |
| **Date:** | |

Aim:

      Write a Java program that reads a file name from the user, displays information about whether the file exists, whether the file is readable, or writable, the type of file and the length of the file in bytes.

Algorithm:

Step 1 Start the Process

Step 2 Prompt the user to enter the file name with path

Step 3 Get the file name

      Step 3.1 Check the file is exists

      Step 3.2 If file exists then proceed to step 3.3 else proceed to step 3.8

      Step 3.3 Check the File is Both Readable and Writeable

      Step 3.4 If yes display file is "Read and Writeable"

      Step 3.5 Else check is readable if yes display "Read Only" else move to step 3.6

      Step 3.6 Else check is writeable if yes display "Write Only"

      Step 3.7 Compute file size and display

      Step 3.8 If file not existing then display "File Not Found"

Step 4 Stop the Process.

Program:
```java
import java.util.Scanner;
import  java.io.File;
class FileDemo
{
        public static void main(String[] args)
        {
                Scanner input=new Scanner(System.in);
                String s=input.nextLine();
                File f1=new File(s);
                System.out.println("File Name:"+f1.getName());
                System.out.println("Path:"+f1.getPath());
                System.out.println("Abs Path:"+f1.getAbsolutePath());
                System.out.println("Parent:"+f1.getParent());
                System.out.println("This file is:"+(f1.exists()?"Exists":"Does not exists"));
                System.out.println("Is file:"+f1.isFile());
                System.out.println("Is Directory:"+f1.isDirectory());
                System.out.println("Is Readable:"+f1.canRead());
                System.out.println("IS Writable:"+f1.canWrite());
                System.out.println("Is Absolute:"+f1.isAbsolute());
                System.out.println("File Last Modified:"+f1.lastModified());
                System.out.println("File Size:"+f1.length()+"bytes");
                System.out.println("Is Hidden:"+f1.isHidden());
        }
}
```

Output:

```
D:\>java FileDemo
D:\testin.txt
File Name:testin.txt
Path:D:\testin.txt
Abs Path:D:\testin.txt
Parent:D:\
This file is:Exists
Is file:true
Is Directory:false
Is Readable:true
IS Writable:true
Is Absolute:true
File Last Modified:1535693725944
File Size:11bytes
Is Hidden:false

D:\>
```

Result:

The java console application for handling files was developed and tested successfully.

Ex.No: 12	**Java Application for Multi threading**
Date:	

Aim:

Write a java program that implements a multi-threaded application that has three threads. First thread generates a random integer every 1 second and if the value is even, second thread computes the square of the number and prints. If the value is odd, the third thread will print the value of cube of the number.

Algorithm:

Step 1 Start the Process

Step 2 Create a thread that generates random number

Step 3 Obtain one random number and check is odd or even

       Step 3.1 If number is even then create and start thread that computes square of a number

       Step 3.2 Compute number * number and display the answer

       Step 3.3 Notify to Random number thread and goto step 4

       Step 3.4 If number is odd then create and start thread that computes cube of a number

       Step 3.5 Compute number * number * number and display the answer

       Step 3.6 Notify to Random number thread and goto step 4

Step 4 Wait for 1 Second and Continue to Step 3 until user wants to exits.
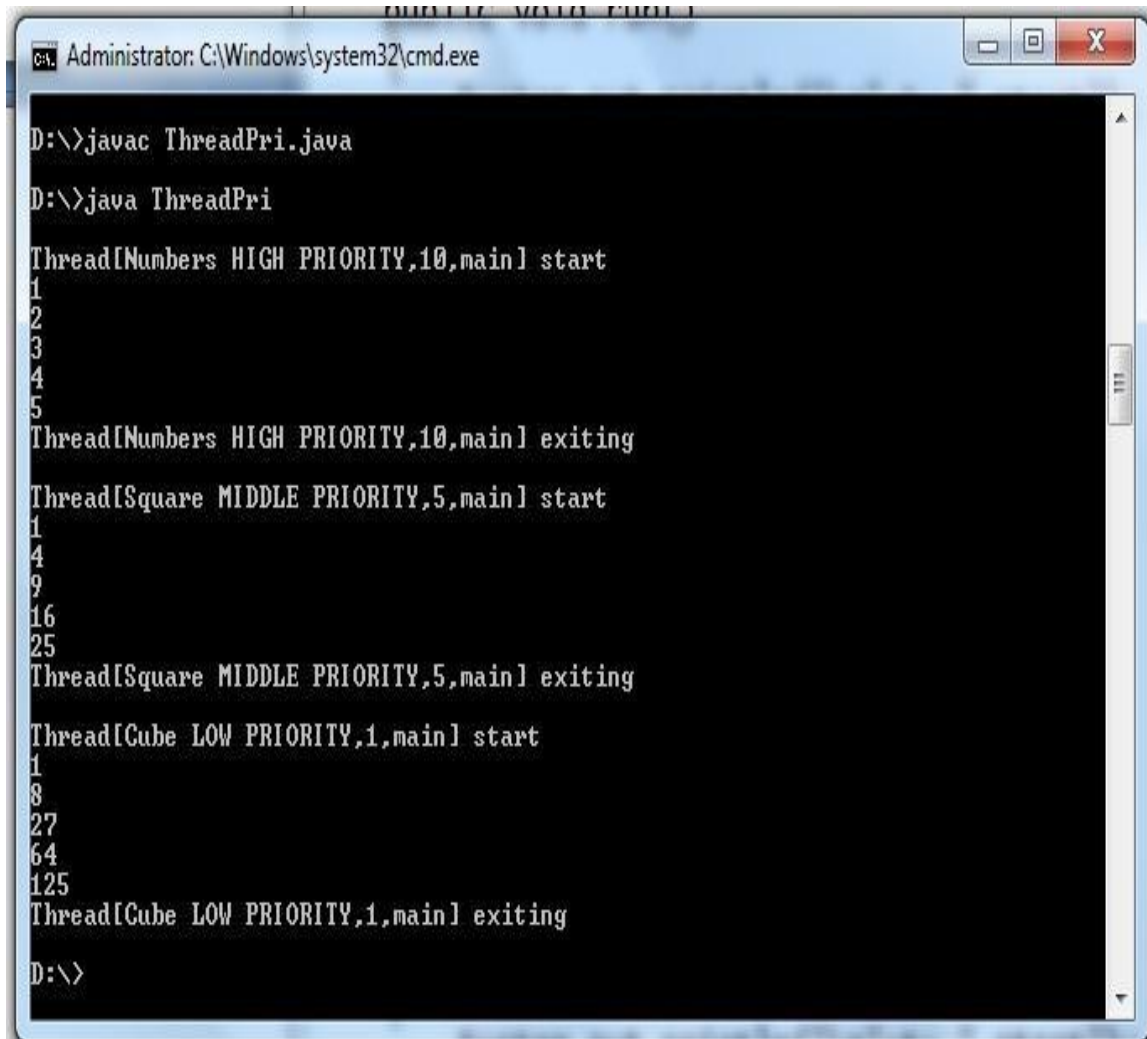
Step 5 Stop the Process

Program:

```java
import java.io.*;
import java.util.*;
class numbers implements Runnable
{
    Thread t;
    boolean running=true;
    public numbers(String name, int p)
    {
        t=new Thread(this,name);
        t.setPriority(p);
        t.start();
    }
    publicvoid run()
    {
        System.out.println("\n"+t+ " start");
        for(int i=1;i<=5;i++)
        {
            System.out.println(i);
        }
        System.out.println(t+ " exiting");
    }
}
class squareRoot implements Runnable
{
    Thread t;
    boolean running=true;
    public squareRoot(String name,int p)
    {
        t=new Thread(this,name);
        t.setPriority(p);
        t.start();
    }
    publicvoid run()
    {
        System.out.println("\n"+t+ " start");
        for(int i=1;i<=5;i++)
        {
            System.out.println(i*i);
        }
        System.out.println(t+ " exiting");
    }
}
class ThreadPri
{
    publicstaticvoid main(String args[])
    {
        new numbers("Numbers HIGH PRIORITY",Thread.MAX_PRIORITY);
        new squareRoot("Square MIDDLE PRIORITY",Thread.NORM_PRIORITY);
        Thread t=Thread.currentThread();
        t.setPriority(Thread.MIN_PRIORITY);
        t.setName("Cube LOW PRIORITY");

        System.out.println("\n"+t+ " start");
        for(int i=1;i<=5;i++)
        {
            System.out.println(i*i*i);
        }
        System.out.println(t+ " exiting");
    }
}
```

Output:

Result:
The java console application for multithreading was developed and tested successfully.

Ex.No: 13	Java Application for Generic Max Finder
Date:	

Aim:

Write a java program to find the maximum value from the given type of elements using a generic function.

Algorithm:

Step 1 Start the Process

Step 2 Create a array of number and array of strings and pass it to generic function.

Step 3 If the array is Integer type

        Step 3.1 Assume first element as MAX

        Step 3.2 Compare [Numeric Perspective] this element with MAX

        Step 3.3 If it is greater than MAX then store current element as MAX

        Step 3.4 Else do nothing

        Step 3.5 Goto step 3.1 until all the elements has been processed.

Step 4 If the array is String type

        Step 4.1 Assume the first element as MAX

        Step 4.2 Compare [Dictionary Perspective] this element with MAX

        Step 4.3 If it is greater than MAX then store current element as MAX

        Step 4.4 Else do nothing

        Step 4.5 Goto Step 3.1 until all the elements has been processed.

Step 5 Stop the Process

Program:

```java
import java.io.*;
import java.util.*;
public class GenericMax
{
    public static void main(String[] args)
        {
        Integer[] numbers = {1, 2, 3}; //Creates array of integers
        System.out.println(max(numbers));
        String[] words = {"red", "green", "blue"}; //Creates an array of strings
        System.out.println(max(words));
        Circle[] circles = {new Circle(3), new Circle(2.9), new Circle(5.9)}; //creates an array of circles
        System.out.println(max(circles));
        }
    static class Circle implements Comparable<Circle> { //Circle object implements Comparable to compare with
Circle
        double radius;
        public Circle (double radius)
            {
            this.radius = radius; //sets value of radius from main method for radius
            }
        @Override
        public int compareTo(Circle c)
            {
            if (radius < c.radius)
                return -1;
            else if (radius == c.radius)
                return 0;
            else
                return 1;
        }
        @Override
        public String toString()
            {
            return "Circle radius: " + radius;
        }
        }
public static <T extends Comparable<T>> T max(T... elements) {
    T max = elements[0];
    for (T element : elements) {
        if (element.compareTo(max) > 0) {
            max = element;
        }
    }
    return max;
}
}
}
```

Output:


```
D:\>javac GenericMax.java

D:\>java GenericMax
3
red
Circle radius: 5.9

D:\>
```

Result:

The java console application for finding generic max of given elements was developed and tested successfully.

| **Ex.No: 14** | **Java Application for Constructor and Method Display** |
| **Date:** | |

Aim:
Class Media has title and price as data members. Provide constructor and method Display() to print the data members. Class Book is derived from class Media andhas an Integer member pages.Class Tape is inherited from Media and has a float member time. Provide constructor to initialize the respective data members. Provide display() to print The respective data members. From main() method, display the members of all the Classes.

ALGORITHM:
    STEP 1: Initialize the code
    STEP 2: Enter the credentials to be displayed
    STEP 3: Run the code in jdk and output is displayed
    STEP 4: End of execution

PROGRAM:
```
class Media {
 String title;
 double price;
 Media(String title, double price) {
 this.title = title;
 this.price = price;
 }
 void display() {
 System.out.println("Title: " + title);
 System.out.println("Price: " + price);
 }
}
class Book extends Media {
 int pages;
 Book(String title, double price, int pages) {   super(title, price);
 this.pages = pages;
 }
 void display() {
 super.display();
 System.out.println("Pages: " + pages);   }
}
class Tape extends Media {
 float time;
 Tape(String title, double price, float time) {   super(title, price);
 this.time = time;
 }
 void display() {
 super.display();
 System.out.println("Time: " + time);   }
}
class Farm{
public static void main(String[] args) {
 Media media = new Media("Media Title", 9.99);
 media.display();
 Book book = new Book("Book Title", 19.99, 200);
 book.display();
 Tape tape = new Tape("Tape Title", 29.99, 120.5f);
 tape.display();
 }
}
```

OUTPUT:

```
C:\Program Files\Java\jdk1.8.0_202\bin>javac Farm.java

C:\Program Files\Java\jdk1.8.0_202\bin>java Farm
Title: Media Title
Price: 9.99
Title: Book Title
Price: 19.99
Pages: 200
Title: Tape Title
Price: 29.99
Time: 120.5
```

RESULT:

Thus we have executed the program and the expected output is displayed.

| **Ex.No: 15** | **Constructor Overloading** |
| **Date:** | |

AIM:

A class Shape is defined with two data members length and breadth and two overloading Constructors in it and a method calculate() to find the area of the rectangle. Derive a class Box with a data member height and define the constructors using super class constructors Chaining in it. Also, override a method calculate() to find the volume of a box. From Main() method, Call the appropriate constructors and methods.

ALGORITHM:

STEP 1: Initialize the code.

STEP 2: Enter the credentials to be displayed.

STEP 3: Run the code in jdk and output is displayed.

STEP 4: End of execution.

PROGRAM:

```
class Shape {

 double length, breadth;

 Shape(double l, double b) {

 length = l;

 breadth = b;}

 Shape(double s) {

 length = breadth = s;}

 double calculate() {

 return length * breadth;}}

class Box extends Shape {

 double height;

 Box(double l, double b, double h) {

 super(l, b);

 height = h;}

 Box(double s, double h) {
 super(s);

 height = h; }

 double calculate() {

 return length * breadth * height;

 }}
```

```
class Farm {

 public static void main(String[] args) {

 Shape rectangle = new Shape(10.0, 20.0);

 System.out.println("Area of rectangle: " + rectangle.calculate());   Shape square =

new Shape(5.0);

 System.out.println("Area of square: " + square.calculate());   Box box = new

Box(5.0, 10.0, 15.0);

 System.out.println("Volume of box: " + box.calculate());   Box cube = new

Box(5.0, 10.0);

 System.out.println("Volume of cube: " + cube.calculate());   }

 }
```

OUTPUT:

```
C:\Program Files\Java\jdk1.8.0_202\bin>javac Farm.java

C:\Program Files\Java\jdk1.8.0_202\bin>java Farm
Area of rectangle: 200.0
Area of square: 25.0
Volume of box: 750.0
Volume of cube: 250.0
```

RESULT:

Thus we have executed the program and the expected output is displayed.

Ex.No: 16	**JAVA Application for User Defined Exceptions**
Date:	

AIM:

A bank wants to conduct examinations for recruitment. You need to develop an application for the applicants to submit their details by implementing the classes based on the description given below.
Method Description

Validate Name(String name)

Validate that the name is not null or empty. If the name is null or empty, return false, else return true.

Validate JobProfile(String jobProfile)

Validate that the jobProfile is either 'Associate' or 'Clerk' or 'Executive' or 'Officer'. If the jobProfile is valid, return true, else return false. Perform case insensitive comparison

Validate Age(int age)

Validate that the age is between 18 and 30 (both inclusive). If the age is valid, return true, else return false

validate(Applicant applicant)

Validate the details of the applicant by calling the appropriate methods. If any validation fails, throw user defined exceptions based on the below description

field violated - name

user defined exception - InvalidNameException

Exception message - Invalid name.

Field violated	User defined exception	Exception message
name	InvalidNameException	Invalid name
jobProfile	InvalidJobProfileException	Invalid job profile
age	InvalidAgeException	Invalid age

• Create an object of Applicant class and set the values of all the instance variables
• Validate the details of the applicant by invoking the validate() method of the Validator class
• If all the details are valid, display 'Application submitted successfully!', else, display appropriate error message

ALGORITHM:

Step 1: Initialize execution

Step 2: Enter the value of n & the condition will be checked

Step 3: After checking the answer will return either yes or no

Step 4: End of execution

PROGRAM :

```
class InvalidNameException extends Exception {
 public InvalidNameException(String message) {
 super(message);
 } }
```

```java
class Validator {

boolean validateName(String name) throws InvalidNameException {   if (name == null ||
name.isEmpty()) {
throw new InvalidNameException("Invalid name.");

 }

 return true;

 }

 boolean validateJobProfile(String jobProfile) {

 String[] validProfiles = {"Associate", "Clerk", "Executive", "Officer"};   for (String profile :
validProfiles) {
 if (jobProfile.equalsIgnoreCase(profile)) {

 return true;

 }
 }

 return false;

 }

 boolean validateAge(int age) {

 return age >= 18 && age <= 30;

 }

 void validate(Applicant applicant) throws Exception {   if
(!validateName(applicant.getName())) {   throw new
InvalidNameException("Invalid name.");    }
 if (!validateJobProfile(applicant.getJobProfile())) {   throw new
Exception("Invalid job profile.");    }
 if (!validateAge(applicant.getAge())) {

 throw new Exception("Invalid age.");    }
 }

 }

 class Applicant {

 private String name;

 private String jobProfile;

 private int age;

 public String getName() {

 return name;

 }

 public void setName(String name) {

 this.name = name;
 }
```

```java
    public String getJobProfile() {

    return jobProfile;

    }

    public void setJobProfile(String jobProfile) {

    this.jobProfile = jobProfile;

    }

    public int getAge() {

    return age;

    }

    public void setAge(int age) {

    this.age = age;

    }

    }

public class Tester {

    public static void main(String[] args) {

    Applicant applicant = new Applicant();

    applicant.setName("John Doe");

    applicant.setJobProfile("Clerk");

    applicant.setAge(25);

    Validator validator = new Validator();

    try {

    validator.validate(applicant);

    System.out.println("Application submitted successfully!");    } catch
(InvalidNameException e) {
    System.out.println("Error: " + e.getMessage());    } catch (Exception e) {
    System.out.println("Error: " + e.getMessage());

    }

    }

    }
```

Output :

```
C:\Program Files\Java\jdk1.8.0_202\bin>javac Farm.java

C:\Program Files\Java\jdk1.8.0_202\bin>java Farm
Application submitted successfully!

C:\Program Files\Java\jdk1.8.0_202\bin>javac Farm.java

C:\Program Files\Java\jdk1.8.0_202\bin>java Farm
Error: Invalid name.

C:\Program Files\Java\jdk1.8.0_202\bin>javac Farm.java

C:\Program Files\Java\jdk1.8.0_202\bin>java Farm
Error: Invalid job profile.
```

RESULT:

Thus a java code has been written and the output has been displayed successfully

| **Ex.No: 17** | **JAVA Application using ArrayList** |
| **Date:** | |

Aim :

Write a Java Code with its output as follows:

Create an array list and add the strings such as "C++,Java, C, python" and implement the following operations:

- Retrieve the string at index 2
- Change the string at index position 2 as "programming"
- Remove the string at index position 0

Algorithm :

Step 1 : Initialize execution

Step 2 : Enter the value of n & the condition will be checked

Step 3 : After checking the answer will return either yes or no

Step 4 : End of execution

Program :

```java
import java.util.ArrayList;

public class ArrayListExample {

 public static void main(String[] args) {

 ArrayList<String> programmingLanguages = new ArrayList<>();
programmingLanguages.add("C++");
 programmingLanguages.add("Java");

 programmingLanguages.add("C");

 programmingLanguages.add("python");

 String languageAtIndex2 = programmingLanguages.get(2);   System.out.println("String
at index 2: " + languageAtIndex2);   // Change the string at index position 2 to
"programming"   programmingLanguages.set(2, "programming");
 System.out.println("Array list after changing element at index 2: " +  programmingLanguages);
 // Remove the string at index position 0

 programmingLanguages.remove(0);

 System.out.println("Array list after removing element at index 0: " +
programmingLanguages);

 }

 }
```

Output :

```
C:\Program Files\Java\jdk1.8.0_202\bin>javac Farm.java

C:\Program Files\Java\jdk1.8.0_202\bin>java Farm
String at index 2: C
Array list after changing element at index 2: [C++, Java, programming, python]
Array list after removing element at index 0: [Java, programming, python]
```

Result :

       Thus a java code has been written and the output has been displayed successfully.

Aim :

Write java Code as follows.

• Create a class "Employee" with data members "empName","empAge","empSalary". • Add appropriate get/set methods for the data members.

• Create an Exception class "EmpSalaryException" with an appropriate constructor. • Create a class EmployeeService with the main() method.

• Write a method checkEmployeeSalary(Employee emp) in EmployeeService, which checks if empSalary<1000 then throws EmpSalaryException.

• In the EmployeeService class main() method

• Create five objects of employee class by passing appropriate values to all the data members.

• Call checkEmployeeSalary() method for each of the five Employee objects being passed and handle the EmpSalaryException.

• Print the empName where the Salary is < 1000.

Algorithm :

Step 1 : Initialize execution

Step 2 : Enter the value of n & the condition will be checked

Step 3 : After checking the answer will return either yes or no

Step 4 : End of execution

Program :

```
public class Employee {
 private String empName;
 private int empAge;
 private double empSalary;
 public String getEmpName() {
 return empName;
 }
 public void setEmpName(String empName) {
 this.empName = empName;
 }
 public int getEmpAge() {
 return empAge; }
 public void setEmpAge(int empAge) {
 this.empAge = empAge;
 }
 public double getEmpSalary() {
 return empSalary;
 }
```

```java
    public void setEmpSalary(double empSalary) {

    this.empSalary = empSalary;

    }}

class EmpSalaryException extends Exception {

    public EmpSalaryException(String message) {

    super(message);

    }}

public class EmployeeService {

    public static void checkEmployeeSalary(Employee emp) throws  EmpSalaryException
{

    if (emp.getEmpSalary() < 1000) {

    throw new EmpSalaryException("Employee salary is less than 1000");   } }
    public static void main(String[] args) {

    Employee emp1 = new Employee();

    emp1.setEmpName("John");

    emp1.setEmpAge(25);

    emp1.setEmpSalary(1500);

    Employee emp2 = new Employee();

    emp2.setEmpName("Bob");

    emp2.setEmpAge(30);

    emp2.setEmpSalary(800);
    Employee emp3 = new Employee();

    emp3.setEmpName("Alice");

    emp3.setEmpAge(28);

    emp3.setEmpSalary(1200);

    Employee emp4 = new Employee();

    emp4.setEmpName("Mark");

    emp4.setEmpAge(27);

    emp4.setEmpSalary(900);

    Employee emp5 = new Employee();

    emp5.setEmpName("Sarah");

    emp5.setEmpAge(32);

    emp5.setEmpSalary(2000);

    Employee[] employees = { emp1, emp2, emp3, emp4, emp5 };   for (Employee emp :
employees) {
    try {
```

```
checkEmployeeSalary(emp);

} catch (EmpSalaryException e) {

System.out.println(emp.getEmpName() + ": " + e.getMessage());   }}}}
```

Output :

```
C:\Program Files\Java\jdk1.8.0_202\bin>javac Farm.java

C:\Program Files\Java\jdk1.8.0_202\bin>java Farm
Bob: Employee salary is less than 1000
Mark: Employee salary is less than 1000
```

Result :

Thus a java code has been written and the output has been displayed successfully.

Ex.No: 19	**Calculator Application using java AWT packages**
Date:	

Aim:

Design a calculator using event-driven programming paradigm of Java with the following options.
        a) Decimal manipulations
        b) Scientific manipulations

Algorithm:

Step 1 Start the Process

Step 2 Display Text View and Number Pad and Option Pads

Step 3 If user presses any number get the existing numbers in Text View add them up and display

Step 4 If user press Operators

    Step 4.1 Get the Text View content as Operant 1 and Set the display to null.

    Step 4.2 If user pressed "+" button set operator as plus

    Step 4.3 If user pressed "-" button set operator as minus

    Step 4.4 If user pressed "x" button set operator as multiply

    Step 4.5 If user pressed "/" button set operator as divide

    Step 4.6 Goto step 3

Step 5 If user pressed "=" button then proceed following steps.

    Step 5.1 Get the Text View content as Operant 2 and Set the display to null.

    Step 5.2 If operator is "plus" then display Text View as Operant 1 + Operant 2

    Step 5.3 If operator is "minus" then display Text View as Operant 1 - Operant 2

    Step 5.4 If operator is "multiply" then display Text View as Operant 1 * Operant 2

    Step 5.5 If operator is "divide" then display Text View as Operant 1 / Operant 2

    Step 5.6 Goto step 4

Step 6 If advanced button pressed

    Step 6.1 Change Operant Types [+,-,x,/ into sin, cos, tan, log] and goto step 2

Step 7 If user pressed any of the operator

    Step 7.1 Get the Text View content as Operant 1 and Set the display to null.

    Step 7.2 If user pressed "sin" button set display the sin value of Operant 1

    Step 7.3 If user pressed "cos" button set display the cos value of Operant 1

    Step 7.4 If user pressed "tan" button set display the tan value of Operant 1

    Step 7.5 If user pressed "log" button set display the log value of Operant 1

    Step 7.6 Goto step 7

Step 8 If advanced pressed again then revert the button changes and return back to normal

Step 9 Repeat the process until user presses the close button then Stop the Process.

Program:

```java
import java.awt.*;
import javax.swing.*;
import java.awt.event.*;
import javax.swing.event.*;
public class ScientificCalculator extends JFrame implements ActionListener {
        JTextField tfield;
        double temp, temp1, result, a;
        static double m1, m2;
        int k = 1, x = 0, y = 0, z = 0;
        char ch;
        JButton b1, b2, b3, b4, b5, b6, b7, b8, b9, zero, clr, pow2, pow3, exp,fac, plus, min, div, log, rec, mul,
eq, addSub, dot, mr, mc, mp,mm, sqrt, sin, cos, tan;
        Container cont;
        JPanel textPanel, buttonpanel;
        ScientificCalculator() {
                cont = getContentPane();
                cont.setLayout(new BorderLayout());
                JPanel textpanel = new JPanel();
                tfield = new JTextField(25);
                tfield.setHorizontalAlignment(SwingConstants.RIGHT);
                tfield.addKeyListener(new KeyAdapter() {
                        public void keyTyped(KeyEvent keyevent) {
                                char c = keyevent.getKeyChar();
                                if (c >= '0' && c <= '9') {
                                } else {
                                        keyevent.consume();
                                }
                        }
                });
                textpanel.add(tfield);
                buttonpanel = new JPanel();
                buttonpanel.setLayout(new GridLayout(8, 4, 2, 2));
                boolean t = true;
                mr = new JButton("MR");
                buttonpanel.add(mr);
                mr.addActionListener(this);
                mc = new JButton("MC");
                buttonpanel.add(mc);
                mc.addActionListener(this);
                mp = new JButton("M+");
                buttonpanel.add(mp);
                mp.addActionListener(this);
                mm = new JButton("M-");
                buttonpanel.add(mm);
                mm.addActionListener(this);
                b1 = new JButton("1");
                buttonpanel.add(b1);
                b1.addActionListener(this);
                b2 = new JButton("2");
                buttonpanel.add(b2);
                b2.addActionListener(this);
                b3 = new JButton("3");
                buttonpanel.add(b3);
                b3.addActionListener(this);
                b4 = new JButton("4");
                buttonpanel.add(b4);
                b4.addActionListener(this);
                b5 = new JButton("5");
                buttonpanel.add(b5);
                b5.addActionListener(this);
                b6 = new JButton("6");
                buttonpanel.add(b6);
```

```java
b6.addActionListener(this);
b7 = new JButton("7");
buttonpanel.add(b7);
b7.addActionListener(this);
b8 = new JButton("8");
buttonpanel.add(b8);
b8.addActionListener(this);
b9 = new JButton("9");
buttonpanel.add(b9);
b9.addActionListener(this);
zero = new JButton("0");
buttonpanel.add(zero);
zero.addActionListener(this);
plus = new JButton("+");
buttonpanel.add(plus);
plus.addActionListener(this);
min = new JButton("-");
buttonpanel.add(min);
min.addActionListener(this);
mul = new JButton("*");
buttonpanel.add(mul);
mul.addActionListener(this);
div = new JButton("/");
div.addActionListener(this);
buttonpanel.add(div);
addSub = new JButton("+/-");
buttonpanel.add(addSub);
addSub.addActionListener(this);
dot = new JButton(".");
buttonpanel.add(dot);
dot.addActionListener(this);
eq = new JButton("=");
buttonpanel.add(eq);
eq.addActionListener(this);
rec = new JButton("1/x");
buttonpanel.add(rec);
rec.addActionListener(this);
sqrt = new JButton("Sqrt");
buttonpanel.add(sqrt);
sqrt.addActionListener(this);
log = new JButton("log");
buttonpanel.add(log);
log.addActionListener(this);
sin = new JButton("SIN");
buttonpanel.add(sin);
sin.addActionListener(this);
cos = new JButton("COS");
buttonpanel.add(cos);
cos.addActionListener(this);
tan = new JButton("TAN");
buttonpanel.add(tan);
tan.addActionListener(this);
pow2 = new JButton("x^2");
buttonpanel.add(pow2);
pow2.addActionListener(this);
pow3 = new JButton("x^3");
buttonpanel.add(pow3);
pow3.addActionListener(this);
exp = new JButton("Exp");
exp.addActionListener(this);
buttonpanel.add(exp);
fac = new JButton("n!");
fac.addActionListener(this);
```

```java
                    buttonpanel.add(fac);
                    clr = new JButton("AC");
                    buttonpanel.add(clr);
                    clr.addActionListener(this);
                    cont.add("Center", buttonpanel);
                    cont.add("North", textpanel);
                    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        }
        public void actionPerformed(ActionEvent e) {
                    String s = e.getActionCommand();
                    if (s.equals("1")) {
                            if (z == 0) {
                                    tfield.setText(tfield.getText() + "1");
                            } else {
                                    tfield.setText("");
                                    tfield.setText(tfield.getText() + "1");
                                    z = 0;
                            }
                    }
                    if (s.equals("2")) {
                            if (z == 0) {
                                    tfield.setText(tfield.getText() + "2");
                            } else {
                                    tfield.setText("");
                                    tfield.setText(tfield.getText() + "2");
                                    z = 0;
                            }
                    }
                    if (s.equals("3")) {
                            if (z == 0) {
                                    tfield.setText(tfield.getText() + "3");
                            } else {
                                    tfield.setText("");
                                    tfield.setText(tfield.getText() + "3");
                                    z = 0;
                            }
                    }
                    if (s.equals("4")) {
                            if (z == 0) {
                                    tfield.setText(tfield.getText() + "4");
                            } else {
                                    tfield.setText("");
                                    tfield.setText(tfield.getText() + "4");
                                    z = 0;
                            }
                    }
                    if (s.equals("5")) {
                            if (z == 0) {
                                    tfield.setText(tfield.getText() + "5");
                            } else {
                                    tfield.setText("");
                                    tfield.setText(tfield.getText() + "5");
                                    z = 0;
                            }
                    }
                    if (s.equals("6")) {
                            if (z == 0) {
                                    tfield.setText(tfield.getText() + "6");
                            } else {
                                    tfield.setText("");
                                    tfield.setText(tfield.getText() + "6");
                                    z = 0;
                            }
```

```java
                }
                if (s.equals("7")) {
                        if (z == 0) {
                                tfield.setText(tfield.getText() + "7");
                        } else {
                                tfield.setText("");
                                tfield.setText(tfield.getText() + "7");
                                z = 0;
                        }
                }
                if (s.equals("8")) {
                        if (z == 0) {
                                tfield.setText(tfield.getText() + "8");
                        } else {
                                tfield.setText("");
                                tfield.setText(tfield.getText() + "8");
                                z = 0;
                        }
                }
                if (s.equals("9")) {
                        if (z == 0) {
                                tfield.setText(tfield.getText() + "9");
                        } else {
                                tfield.setText("");
                                tfield.setText(tfield.getText() + "9");
                                z = 0;
                        }
                }
                if (s.equals("0")) {
                        if (z == 0) {
                                tfield.setText(tfield.getText() + "0");
                        } else {
                                tfield.setText("");
                                tfield.setText(tfield.getText() + "0");
                                z = 0;
                        }
                }
                if (s.equals("AC")) {
                        tfield.setText("");
                        x = 0;
                        y = 0;
                        z = 0;
                }
                if (s.equals("log")) {
                        if (tfield.getText().equals("")) {
                                tfield.setText("");
                        } else {
                                a = Math.log(Double.parseDouble(tfield.getText()));
                                tfield.setText("");
                                tfield.setText(tfield.getText() + a);
                        }
                }
                if (s.equals("1/x")) {
                        if (tfield.getText().equals("")) {
                                tfield.setText("");
                        } else {
                                a = 1 / Double.parseDouble(tfield.getText());
                                tfield.setText("");
                                tfield.setText(tfield.getText() + a);
                        }
                }
                if (s.equals("Exp")) {
                        if (tfield.getText().equals("")) {
```

```java
                                tfield.setText("");
                        } else {
                                a = Math.exp(Double.parseDouble(tfield.getText()));
                                tfield.setText("");
                                tfield.setText(tfield.getText() + a);
                        }
                }
                if (s.equals("x^2")) {
                        if (tfield.getText().equals("")) {
                                tfield.setText("");
                        } else {
                                a = Math.pow(Double.parseDouble(tfield.getText()), 2);
                                tfield.setText("");
                                tfield.setText(tfield.getText() + a);
                        }
                }
                if (s.equals("x^3")) {
                        if (tfield.getText().equals("")) {
                                tfield.setText("");
                        } else {
                                a = Math.pow(Double.parseDouble(tfield.getText()), 3);
                                tfield.setText("");
                                tfield.setText(tfield.getText() + a);
                        }
                }
                if (s.equals("+/-")) {
                        if (x == 0) {
                                tfield.setText("-" + tfield.getText());
                                x = 1;
                        } else {
                                tfield.setText(tfield.getText());
                        }
                }
                if (s.equals(".")) {
                        if (y == 0) {
                                tfield.setText(tfield.getText() + ".");
                                y = 1;
                        } else {
                                tfield.setText(tfield.getText());
                        }
                }
                if (s.equals("+")) {
                        if (tfield.getText().equals("")) {
                                tfield.setText("");
                                temp = 0;
                                ch = '+';
                        } else {
                                temp = Double.parseDouble(tfield.getText());
                                tfield.setText("");
                                ch = '+';
                                y = 0;
                                x = 0;
                        }
                        tfield.requestFocus();
                }
                if (s.equals("-")) {
                        if (tfield.getText().equals("")) {
                                tfield.setText("");
                                temp = 0;
                                ch = '-';
                        } else {
                                x = 0;
                                y = 0;
```

```java
                                temp = Double.parseDouble(tfield.getText());
                                tfield.setText("");
                                ch = '-';
                        }
                        tfield.requestFocus();
                }
                if (s.equals("/")) {
                        if (tfield.getText().equals("")) {
                                tfield.setText("");
                                temp = 1;
                                ch = '/';
                        } else {
                                x = 0;
                                y = 0;
                                temp = Double.parseDouble(tfield.getText());
                                ch = '/';
                                tfield.setText("");
                        }
                        tfield.requestFocus();
                }
                if (s.equals("*")) {
                        if (tfield.getText().equals("")) {
                                tfield.setText("");
                                temp = 1;
                                ch = '*';
                        } else {
                                x = 0;
                                y = 0;
                                temp = Double.parseDouble(tfield.getText());
                                ch = '*';
                                tfield.setText("");
                        }
                        tfield.requestFocus();
                }
                if (s.equals("MC")) {
                        m1 = 0;
                        tfield.setText("");
                }
                if (s.equals("MR")) {
                        tfield.setText("");
                        tfield.setText(tfield.getText() + m1);
                }
                if (s.equals("M+")) {
                        if (k == 1) {
                                m1 = Double.parseDouble(tfield.getText());
                                k++;
                        } else {
                                m1 += Double.parseDouble(tfield.getText());
                                tfield.setText("" + m1);
                        }
                }
                if (s.equals("M-")) {
                        if (k == 1) {
                                m1 = Double.parseDouble(tfield.getText());
                                k++;
                        } else {
                                m1 -= Double.parseDouble(tfield.getText());
                                tfield.setText("" + m1);
                        }
                }
                if (s.equals("Sqrt")) {
                        if (tfield.getText().equals("")) {
                                tfield.setText("");
```

```java
                } else {
                        a = Math.sqrt(Double.parseDouble(tfield.getText()));
                        tfield.setText("");
                        tfield.setText(tfield.getText() + a);
                }
        }
        if (s.equals("SIN")) {
                if (tfield.getText().equals("")) {
                        tfield.setText("");
                } else {
                        a = Math.sin(Double.parseDouble(tfield.getText()));
                        tfield.setText("");
                        tfield.setText(tfield.getText() + a);
                }
        }
        if (s.equals("COS")) {
                if (tfield.getText().equals("")) {
                        tfield.setText("");
                } else {
                        a = Math.cos(Double.parseDouble(tfield.getText()));
                        tfield.setText("");
                        tfield.setText(tfield.getText() + a);
                }
        }
        if (s.equals("TAN")) {
                if (tfield.getText().equals("")) {
                        tfield.setText("");
                } else {
                        a = Math.tan(Double.parseDouble(tfield.getText()));
                        tfield.setText("");
                        tfield.setText(tfield.getText() + a);
                }
        }
        if (s.equals("=")) {
                if (tfield.getText().equals("")) {
                        tfield.setText("");
                } else {
                        temp1 = Double.parseDouble(tfield.getText());
                        switch (ch) {
                        case '+':
                                result = temp + temp1;
                                break;
                        case '-':
                                result = temp - temp1;
                                break;
                        case '/':
                                result = temp / temp1;
                                break;
                        case '*':
                                result = temp * temp1;
                                break;
                        }
                        tfield.setText("");
                        tfield.setText(tfield.getText() + result);
                        z = 1;
                }
        }
        if (s.equals("n!")) {
                if (tfield.getText().equals("")) {
                        tfield.setText("");
                } else {
                        a = fact(Double.parseDouble(tfield.getText()));
                        tfield.setText("");
```

```java
                                tfield.setText(tfield.getText() + a);
                        }
                }
                tfield.requestFocus();
        }
        double fact(double x) {
                int er = 0;
                if (x < 0) {
                        er = 20;
                        return 0;
                }
                double i, s = 1;
                for (i = 2; i <= x; i += 1.0)
                        s *= i;
                return s;
        }
        public static void main(String args[]) {
                try {
                UIManager.setLookAndFeel("com.sun.java.swing.plaf.windows.WindowsLookAndFeel");
                } catch (Exception e) {
                }
                ScientificCalculator f = new ScientificCalculator();
                f.setTitle("ScientificCalculator");
                f.pack();
                f.setVisible(true);
        }
}
```

Output:
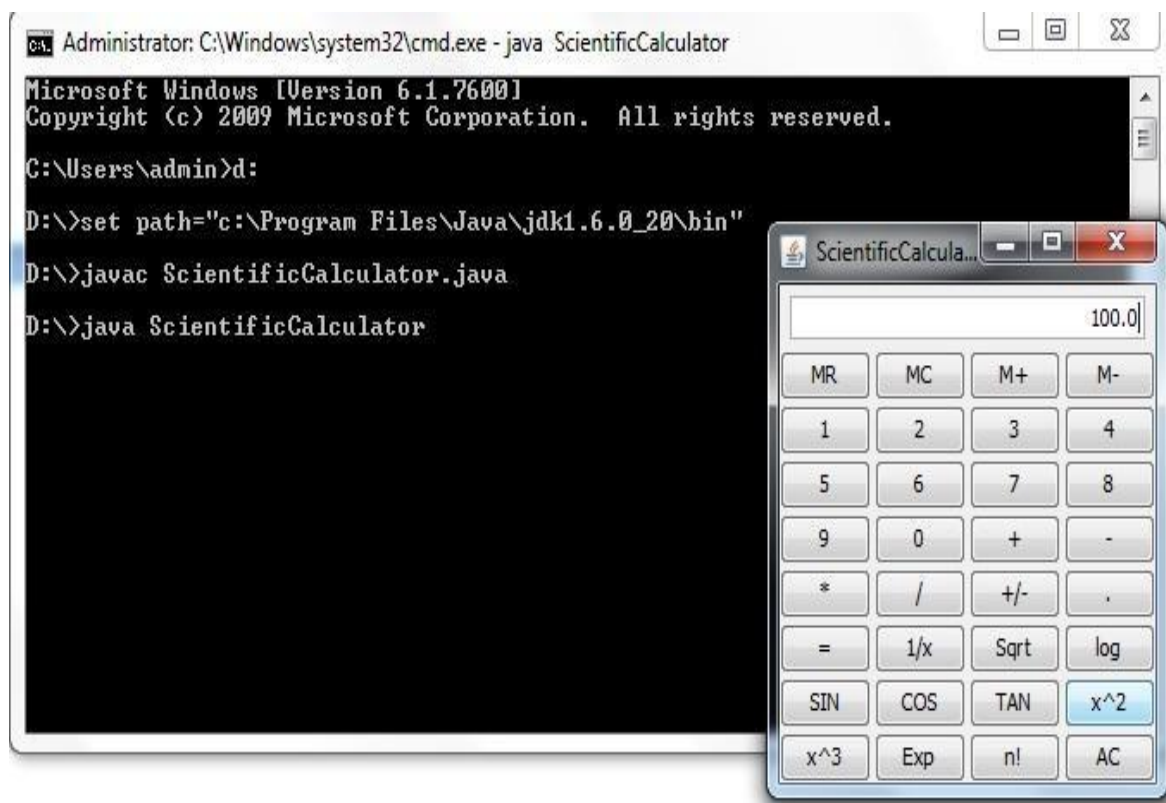
Result:

The java GUI application for calculator with basic and advanced functionalities was developed and tested successfully.

Ex.No: 20 **Date:**	**Simple TEXT EDITOR**

Aim:

Develop a mini project for any application using Java concepts.

Algorithm:

Step 1 Start the Process

Step 2 Display Text View and Number Pad and Option Pads

Step 3 Using the swing components design the necessary layout

Step 4 Create necessary menu's as you want.

Step 5 Stop the process once you complete.

Program: (SIMPLE TEXT EDITOR)

```java
//THE IMPORTED LIBRARIES
import javax.swing.*;
import javax.swing.undo.*;
import java.io.*;
import java.awt.*;
import java.awt.event.*;
import javax.swing.event.*;
import java.util.*;
import javax.swing.text.*;
import java.text.*;
public class myedit extends JFrame implements ActionListener
{
        public static myedit e;
        //DECLARATION OF ALL THE VARIABLES USED IN THIS APPLICATION
        JTextArea text = new JTextArea(0,0);
        JScrollPane scroll = new JScrollPane(text);
        JMenuBar mb=new JMenuBar();
        JMenu FILE = new JMenu("File");
        JMenu EDIT = new JMenu("Edit");
        JMenu SEARCH = new JMenu("Search");
        JMenu HELP = new JMenu("Help");
        JMenuItem NEWFILE = new JMenuItem("New");
        JMenuItem OPENFILE = new JMenuItem("Open...");
        JMenuItem SAVEFILE = new JMenuItem("Save");
        JMenuItem SAVEASFILE = new JMenuItem("Save As...");
        JMenuItem EXITFILE = new JMenuItem("Exit");
        JMenuItem CUTEDIT = new JMenuItem("Cut");
        JMenuItem COPYEDIT = new JMenuItem("Copy");
        JMenuItem PASTEDIT = new JMenuItem("Paste");
        JMenuItem DELETEDIT = new JMenuItem("Delete");
        JMenuItem SELECTEDIT = new JMenuItem("Select All");
        JMenuItem TIMEDIT = new JMenuItem("Time/Date");
        JCheckBoxMenuItem WORDEDIT = new JCheckBoxMenuItem("Word Wrap");
        JMenuItem FONTEDIT = new JMenuItem("Set Font...");
        JMenuItem FINDSEARCH = new JMenuItem("Find");
        JMenuItem FINDNEXTSEARCH = new JMenuItem("Find Next");
        JMenuItem ABOUTHELP = new JMenuItem("About");
        JPopupMenu POPUP = new JPopupMenu();
        JMenuItem CUTPOPUP = new JMenuItem("Cut");
        JMenuItem COPYPOPUP = new JMenuItem("Copy");
        JMenuItem PASTEPOPUP = new JMenuItem("Paste");
        JMenuItem DELETEPOPUP = new JMenuItem("Delete");
        JMenuItem SELECTPOPUP = new JMenuItem("Select All");
        UndoManager undo = new UndoManager();
        UndoAction undoAction = new UndoAction();
        boolean opened = false;
        String wholeText,findString,filename = null;
        int ind = 0;
        //CLASS FOR HANDLING UNDO MENU OPTION OF EDIT AND POPUP MENU
        class UndoAction extends AbstractAction
        {
                public UndoAction()
                {
                        super("Undo");
                }

                public void actionPerformed(ActionEvent e)
                {
                        try
                        {
                                undo.undo();
                        }
```

```java
                catch (CannotUndoException ex)
                {
                        System.out.println("Unable to undo: " + ex);
                        ex.printStackTrace();
                }
                update();
        }

        protected void update()
        {
                if(undo.canUndo())
                {
                        setEnabled(true);
                        putValue("Undo", undo.getUndoPresentationName());
                }
                else
                {
                        setEnabled(false);
                        putValue(Action.NAME, "Undo");
                }
        }
}
//DEFAULT CONSTRUCTOR OF THE MYEDIT CLASS
public myedit()
{
        //SETING DEFAULT TITLE OF THE FRAME
        setTitle("Untitled");

        //SETTING DEFAULT SIZE OF THE FRAME
        setSize(600,400);

        //MAKING THE FRAME VISIBLE
        setVisible(true);

        //SETTING WORD WRAP TO TRUE AS DEFAULT
        text.setLineWrap(true);

        //SETTING THE DEFAULT STATE OF WORDWRAP MENU OPTION IN EDIT MENU
        WORDEDIT.setState(true);

        //SETTING THE LAYOUT OF THE FRAME
        getContentPane().setLayout(new BorderLayout());

        //ADDS THE SCROLLPANE CONTAINING THE TEXTAREA TO THE CONTAINER
        getContentPane().add(scroll, BorderLayout.CENTER);

        //ADDING THE MAIN MENUBAR TO THE FRAME
        setJMenuBar(mb);

        //ADDING MENUS TO THE MAIN MENUBAR
        mb.add(FILE);
        mb.add(EDIT);
        mb.add(SEARCH);
        mb.add(HELP);

        //ADDING MENUITEMS TO THE FILE MENU
        FILE.add(NEWFILE);
        FILE.add(OPENFILE);
        FILE.add(SAVEFILE);
        FILE.add(SAVEASFILE);
        FILE.addSeparator();
        FILE.add(EXITFILE);
```

```java
//ADDING MENUITEMS TO THE EDIT MENU
EDIT.add(undoAction);
EDIT.add(CUTEDIT);
EDIT.add(COPYEDIT);
EDIT.add(PASTEDIT);
EDIT.add(DELETEDIT);
EDIT.addSeparator();
EDIT.add(SELECTEDIT);
EDIT.add(TIMEDIT);
EDIT.addSeparator();
EDIT.add(WORDEDIT);
EDIT.add(FONTEDIT);

//ADDING MENUITEMS TO THE SEARCH MENU
SEARCH.add(FINDSEARCH);
SEARCH.add(FINDNEXTSEARCH);

//ADDING MENUITEM TO THE HELP MENU
HELP.add(ABOUTHELP);

//ADDING MENUITEMS TO THE POPUPMENU
POPUP.add(undoAction);
POPUP.addSeparator();
POPUP.add(CUTPOPUP);
POPUP.add(COPYPOPUP);
POPUP.add(PASTEPOPUP);
POPUP.add(DELETEPOPUP);
POPUP.addSeparator();
POPUP.add(SELECTPOPUP);

//SETTING SHORTCUT KEYS OF MENUS IN THE MAIN MENUBAR
FILE.setMnemonic(KeyEvent.VK_F);
EDIT.setMnemonic(KeyEvent.VK_E);
SEARCH.setMnemonic(KeyEvent.VK_S);
HELP.setMnemonic(KeyEvent.VK_H);

//SETTING SHORTCUT KEYS OF MENUITEMS IN THE FILE MENU
NEWFILE.setMnemonic(KeyEvent.VK_N);
OPENFILE.setMnemonic(KeyEvent.VK_O);
SAVEFILE.setMnemonic(KeyEvent.VK_S);
SAVEASFILE.setMnemonic(KeyEvent.VK_A);
EXITFILE.setMnemonic(KeyEvent.VK_X);

//SETTING SHORTCUT KEYS OF MENUITEMS IN THE EDIT MENU
CUTEDIT.setMnemonic(KeyEvent.VK_T);
COPYEDIT.setMnemonic(KeyEvent.VK_C);
PASTEDIT.setMnemonic(KeyEvent.VK_P);
DELETEDIT.setMnemonic(KeyEvent.VK_L);
SELECTEDIT.setMnemonic(KeyEvent.VK_A);
TIMEDIT.setMnemonic(KeyEvent.VK_D);
WORDEDIT.setMnemonic(KeyEvent.VK_W);
FONTEDIT.setMnemonic(KeyEvent.VK_F);

//SETTING SHORTCUT KEYS OF MENUITEMS IN THE SEARCH MENU
FINDSEARCH.setMnemonic(KeyEvent.VK_F);
FINDNEXTSEARCH.setMnemonic(KeyEvent.VK_N);

//SETTING SHORTCUT KEYS OF MENUITEM IN THE HELP MENU
ABOUTHELP.setMnemonic(KeyEvent.VK_A);

//SETTING SHORTCUT KEYS OF MENUITEMS IN THE POPUPMENU
CUTPOPUP.setMnemonic(KeyEvent.VK_T);
COPYPOPUP.setMnemonic(KeyEvent.VK_C);
```

```java
        PASTEPOPUP.setMnemonic(KeyEvent.VK_P);
        DELETEPOPUP.setMnemonic(KeyEvent.VK_D);
        SELECTPOPUP.setMnemonic(KeyEvent.VK_A);

        //SETTING ACCELERATOR KEYS OF SOME MENUITEMS IN THE EDIT MENU

CUTEDIT.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_X,ActionEvent.CTRL_MASK));

COPYEDIT.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_C,ActionEvent.CTRL_MASK));

PASTEDIT.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_V,ActionEvent.CTRL_MASK));

        //ADDING LISTENERS TO THE MENUITEMS IN FILE MENU
        NEWFILE.addActionListener(this);
        OPENFILE.addActionListener(this);
        SAVEFILE.addActionListener(this);
        SAVEASFILE.addActionListener(this);
        EXITFILE.addActionListener(this);

        //ADDING LISTENERS TO THE MENUITEMS IN EDIT MENU
        text.getDocument().addUndoableEditListener(new UndoListener());
        CUTEDIT.addActionListener(this);
        COPYEDIT.addActionListener(this);
        PASTEDIT.addActionListener(this);
        DELETEDIT.addActionListener(this);
        SELECTEDIT.addActionListener(this);
        TIMEDIT.addActionListener(this);
        WORDEDIT.addActionListener(this);
        FONTEDIT.addActionListener(this);

        //ADDING LISTENERS TO THE MENUITEMS IN SEARCH MENU
        FINDSEARCH.addActionListener(this);
        FINDNEXTSEARCH.addActionListener(this);

        //ADDING LISTENERS TO THE MENUITEM IN HELP MENU
        ABOUTHELP.addActionListener(this);

        //ADDING LISTENERS TO THE MENUITEMS IN POPUPMENU
        CUTPOPUP.addActionListener(this);
        COPYPOPUP.addActionListener(this);
        PASTEPOPUP.addActionListener(this);
        DELETEPOPUP.addActionListener(this);
        SELECTPOPUP.addActionListener(this);

        //ADDING MOUSELISTENER TO RIGHT CLICK FOR THE POPUPLISTENER
        text.addMouseListener(new MouseAdapter()
        {
          public void mousePressed(MouseEvent e)
          {
                if (e.isPopupTrigger())
                {
              POPUP.show(e.getComponent(),e.getX(), e.getY());
          }

          }
          public void mouseReleased(MouseEvent e)
          {
                        if (e.isPopupTrigger())
                        {
                        POPUP.show(e.getComponent(),e.getX(), e.getY());
              }
          }
        });
```

```java
            //ADDING WINDOWLISTENER TO HANDLE CLOSE WINDOW EVENT

/*      addWindowListener( new WindowAdapter()
        { public void windowClosing(WindowEvent evt)
         {
                int response = JOptionPane.showConfirmDialog(null, "Do you really want to quit?");
                System.out.println("Inside Window Listener");
                switch (response)
                {
                        case 0:
                                {
                                dispose();
                                break; }

                        case 2:
                        {
                                //myedit x = new myedit();
                                System.out.println("Inside 2");
                                e=new myedit();
                                e.setVisible(true);
                                break;}
                 }
                System.out.println("response is :="+response);
         }
        } ); */

        addWindowListener(new WindowAdapter()
        {
                public void windowClosing(WindowEvent e)
                {
                        exitApln();
                }
        });
}

//HANDLING ALL EVENTS OF THE TEXT EDITOR
public void actionPerformed(ActionEvent e)
{

        //ACTION FOR NEW MENU OPTION OF FILE MENU
        if (e.getSource()==NEWFILE)
        {
                newfile();
        }

        //ACTION FOR OPEN MENU OPTION OF FILE MENU
        if (e.getSource()==OPENFILE)
        {
                open();
        }

        //ACTION FOR SAVE MENU OPTION OF FILE MENU
        if (e.getSource()==SAVEFILE)
        {
                save();
        }

        //ACTION FOR SAVEAS MENU OPTION OF FILE MENU
        if (e.getSource()==SAVEASFILE)
        {
                opened=false;
                save();
```

```java
        }

        //ACTION FOR EXIT MENU OPTION OF FILE MENU
        if (e.getSource()==EXITFILE)
        {
                exitApln();
        }

        //ACTION FOR CUT MENU OPTION OF EDIT MENU AND POPUPMENU
        if ((e.getSource()==CUTEDIT)||(e.getSource()==CUTPOPUP))
        {
                text.cut();
        }

        //ACTION FOR COPY MENU OPTION OF EDIT MENU AND POPUPMENU
        if ((e.getSource()==COPYEDIT)||(e.getSource()==COPYPOPUP))
        {
                text.copy();
        }

        //ACTION FOR PASTE MENU OPTION OF EDIT MENU AND POPUPMENU
        if ((e.getSource()==PASTEDIT)||(e.getSource()==PASTEPOPUP))
        {
                text.paste();
        }

        //ACTION FOR DELETE MENU OPTION OF EDIT MENU AND POPUPMENU
        if ((e.getSource()==DELETEDIT)||(e.getSource()==DELETEPOPUP))
        {
                text.replaceSelection(null);
        }

        //ACTION FOR SELECTALL MENU OPTION OF EDIT MENU AND POPUPMENU
        if ((e.getSource()==SELECTEDIT)||(e.getSource()==SELECTPOPUP))
        {
                text.selectAll();
        }

        //ACTION FOR TIME/DATE MENU OPTION OF EDIT MENU
        if (e.getSource()==TIMEDIT)
        {
                Date currDate;
                String dd;
                currDate = new java.util.Date();
                dd=currDate.toString();
                text.insert(dd,text.getCaretPosition());
        }

        //ACTION FOR WORD WRAP MENU OPTION OF EDIT MENU
        if (e.getSource()==WORDEDIT)
        {
                if(WORDEDIT.isSelected())
                text.setLineWrap(true);
                else
                text.setLineWrap(false);
        }

        //ACTION FOR SET FONT MENU OPTION OF EDIT MENU
        if (e.getSource()==FONTEDIT)
        {
                fontDialogBox fontS = new fontDialogBox();
        }
```

```java
                //ACTION FOR FIND MENU OPTION OF SEARCH MENU
                if (e.getSource()==FINDSEARCH)
                {
                        wholeText=text.getText();
                        findString =JOptionPane.showInputDialog(null, "Find What", "Find",
JOptionPane.INFORMATION_MESSAGE);

                        ind = wholeText.indexOf(findString,0);
                        text.setCaretPosition(ind);
                        text.setSelectionStart(ind);
                        text.setSelectionEnd(ind+findString.length());
                }

                //ACTION FOR FIND NEXT MENU OPTION OF SEARCH MENU
                if (e.getSource()==FINDNEXTSEARCH)
                {
                        wholeText= text.getText();
                        findString = JOptionPane.showInputDialog(null, "Find What","Find Next",
JOptionPane.INFORMATION_MESSAGE);
                        ind = wholeText.indexOf(findString, text.getCaretPosition());
                        text.setCaretPosition(ind);
                        text.setSelectionStart(ind);
                        text.setSelectionEnd(ind+findString.length());
                }

                //ACTION FOR ABOUT MENU OPTION OF HELP MENU
                if (e.getSource()==ABOUTHELP)
                {
                        JOptionPane.showMessageDialog(null, "This is a simple Text Editor application built
using Java.",

                        "About Editor",
                        JOptionPane.INFORMATION_MESSAGE);
                }
        }

        //ACTION FOR NEW MENU OPTION OF FILE MENU
        public void newfile()
        {
                if(!text.getText().equals(""))
                {
                        opened=false;
                        int confirm = JOptionPane.showConfirmDialog(null,
                        "Text in the Untitled file has changed. \n Do you want to save the changes?",
                        "New File",

        JOptionPane.YES_NO_CANCEL_OPTION,JOptionPane.INFORMATION_MESSAGE);

                        if( confirm == JOptionPane.YES_OPTION )
                        {
                                save();
                                text.setText(null);
                        }
                        else if( confirm == JOptionPane.NO_OPTION )
                        {
                                text.setText(null);
                        }
                }
        }

        //ACTION FOR OPEN MENU OPTION OF FILE MENU
        public void open()
        {
                text.setText(null);
```

```java
                JFileChooser ch = new JFileChooser();
                ch.setCurrentDirectory(new File("."));
                ch.setFileFilter(new javax.swing.filechooser.FileFilter()
                {
                public boolean accept(File f)
                {
                        return f.isDirectory()
                                || f.getName().toLowerCase().endsWith(".java");
                }
                        public String getDescription()
                {
                    return "Java files";
                }
        });
                int result = ch.showOpenDialog(new JPanel());
                if(result == JFileChooser.APPROVE_OPTION)
                {
                        filename = String.valueOf(ch.getSelectedFile());
                        setTitle(filename);
                        opened=true;
                        FileReader fr;
                        BufferedReader br;

                        try
                        {
                                fr=new FileReader (filename);
                                br=new BufferedReader(fr);
                                String s;
                                while((s=br.readLine())!=null)
                                {
                                        text.append(s);
                                        text.append("\n");
                                }
                                fr.close();
                        }
                        catch(FileNotFoundException ex)
                {
                                JOptionPane.showMessageDialog(this, "Requested file not found", "Error
Dialog box",

JOptionPane.ERROR_MESSAGE);}

                catch(Exception ex)
                  {System.out.println(ex);}
                  }
        }

        //ACTION FOR SAVE MENU OPTION OF FILE MENU
        public void save()
        {
                if(opened==true)
                {
                        try
                        {
                                FileWriter f1 = new FileWriter(filename);
                                f1.write(text.getText());
                                f1.close();
                                opened = true;
                        }
                        catch(FileNotFoundException ex)
                {
```

```java
                                    JOptionPane.showMessageDialog(this, "Requested file not found", "Error
Dialog box",

JOptionPane.ERROR_MESSAGE);}
                        catch(IOException ioe){ioe.printStackTrace();}
                }
                else
                {
                        JFileChooser fc = new JFileChooser();
                        fc.setCurrentDirectory(new File("."));
                        int result = fc.showSaveDialog(new JPanel());

                        if(result == JFileChooser.APPROVE_OPTION)
                        {
                                filename = String.valueOf(fc.getSelectedFile());
                                setTitle(filename);
                                try
                                {
                                FileWriter f1 = new FileWriter(filename);
                                f1.write(text.getText());
                                f1.close();
                                opened = true;
                                }
                        catch(FileNotFoundException ex)
            {
                                    JOptionPane.showMessageDialog(this, "Requested file not found", "Error
Dialog box",

JOptionPane.ERROR_MESSAGE);}

                                catch(IOException ioe){ioe.printStackTrace();}
                        }

                }
        }

        //ACTION FOR EXIT MENU OPTION OF FILE MENU AND CLOSE WINDOW BUTTON
        public void exitApln()
        {
                if(!text.getText().equals(""))
                {
                        int confirm = JOptionPane.showConfirmDialog(null,
                        "Text in the file has changed. \n Do you want to save the changes?",
                        "Exit",

        JOptionPane.YES_NO_CANCEL_OPTION,JOptionPane.INFORMATION_MESSAGE);

                        if( confirm == JOptionPane.YES_OPTION )
                        {
                                save();
                                dispose();
                                System.exit(0);
                        }
                        else if( confirm == JOptionPane.CANCEL_OPTION )
                        {
                                e=new myedit();
                                String s= text.getText();
                                e.setVisible(true);
                                e.text.setText(s);
                        }
                        else if( confirm == JOptionPane.NO_OPTION )
                        {
                                dispose();
```

```java
                            System.exit(0);
                    }
            }
            else
            {
                    System.exit(0);
            }
        }


        //CLASS FOR UNDOLISTENER
        class UndoListener implements UndoableEditListener
        {
                public void undoableEditHappened(UndoableEditEvent e)
                {
                undo.addEdit(e.getEdit());
                undoAction.update();
                }
        }


        //CLASS FOR BUILDING AND DISPLAYING FONT DIALOG BOX
        class fontDialogBox extends JFrame implements ActionListener
        {
                //DECLARATION OF ALL VARIABLES USED IN fontDialogBox CLASS

                String availableFontString[] =
GraphicsEnvironment.getLocalGraphicsEnvironment().getAvailableFontFamilyNames();
                JList fontList = new JList(availableFontString);
                JLabel fontLabel = new JLabel("Font");
                JTextField valueFont=new JTextField("Arial");
                JScrollPane fontPane = new JScrollPane(fontList);


                String fontStyleString[] = {"Normal","Bold","Italic","Bold Italic"};
                JList styleList = new JList(fontStyleString);
                JLabel styleLabel = new JLabel("Style");
                int v = ScrollPaneConstants.VERTICAL_SCROLLBAR_ALWAYS;
                int h = ScrollPaneConstants.HORIZONTAL_SCROLLBAR_AS_NEEDED;
                JScrollPane stylePane = new JScrollPane(styleList,v,h);
                JTextField valueStyle=new JTextField("Normal");

                String fontSizeString[] = {"8","10","12","14","16","18","20","22","24","28"};
                JList sizeList = new JList(fontSizeString);
                JLabel sizeLabel = new JLabel("Font size");
                JScrollPane sizePane = new JScrollPane(sizeList);
                JTextField valueSize=new JTextField("12");

                JButton okButton = new JButton("OK");
                JButton cancelButton = new JButton("Cancel");

                JLabel sampleLabel = new JLabel("Sample:");
                JTextField sample = new JTextField("  AaBbCc");

                Font selectedFont;

                //DEFAULT CONSTRUCTOR OF fontDialogBox CLASS
                public fontDialogBox()
                {
                        setSize(500,300);
                        setTitle("Font");
                        setVisible(true);
                        sample.setEditable(false);

                        getContentPane().setLayout(null);
```

```java
fontLabel.setBounds(10,10,170,20);
valueFont.setBounds(10,35,170,20);
fontPane.setBounds(10,60,170,150);

styleLabel.setBounds(200,10,100,20);
valueStyle.setBounds(200,35,100,20);
stylePane.setBounds(200,60,100,150);

sizeLabel.setBounds(320,10,50,20);
valueSize.setBounds(320,35,50,20);
sizePane.setBounds(320,60,50,150);

okButton.setBounds(400,35,80,20);
cancelButton.setBounds(400,60,80,20);

sampleLabel.setBounds(150,235,50,30);
sample.setBounds(200,235,100,30);

getContentPane().add(fontLabel);
getContentPane().add(fontPane);
getContentPane().add(valueFont);

getContentPane().add(styleLabel);
getContentPane().add(stylePane);
getContentPane().add(valueStyle);

getContentPane().add(sizeLabel);
getContentPane().add(sizePane);
getContentPane().add(valueSize);

getContentPane().add(okButton);
getContentPane().add(cancelButton);
getContentPane().add(sampleLabel);
getContentPane().add(sample);

okButton.addActionListener(this);
cancelButton.addActionListener(this);

fontList.addListSelectionListener(new ListSelectionListener()
{
        public void valueChanged(ListSelectionEvent event)
        {
                if (!event.getValueIsAdjusting())
                {
                valueFont.setText(fontList.getSelectedValue().toString());
                selectedFont = new
Font(valueFont.getText(),styleList.getSelectedIndex
(),Integer.parseInt(valueSize.getText()));
                sample.setFont(selectedFont);
                }
        }
});

styleList.addListSelectionListener(new ListSelectionListener()
{
        public void valueChanged(ListSelectionEvent event)
        {
        if (!event.getValueIsAdjusting())
                {
                valueStyle.setText(styleList.getSelectedValue().toString());
```

```java
                                                selectedFont = new
Font(valueFont.getText(),styleList.getSelectedIndex

(),Integer.parseInt(valueSize.getText()));
                                                sample.setFont(selectedFont);
                                        }
                                }
                        });

                        sizeList.addListSelectionListener(new ListSelectionListener()
                        {
                                public void valueChanged(ListSelectionEvent event)
                                {
                                        if (!event.getValueIsAdjusting())
                                        {
                                                valueSize.setText(sizeList.getSelectedValue().toString());
                                                selectedFont = new
Font(valueFont.getText(),styleList.getSelectedIndex

(),Integer.parseInt(valueSize.getText()));
                                                sample.setFont(selectedFont);
                                        }
                                }
                        });
                }//END OF DEFAULT CONSTRUCTOR OF fontdialogBox CLASS

                public void actionPerformed(ActionEvent ae)
                {
                        if(ae.getSource()==okButton)
                        {
                                selectedFont = new
Font(valueFont.getText(),styleList.getSelectedIndex(),Integer.parseInt

(valueSize.getText()));
                                text.setFont(selectedFont);
                                setVisible(false);
                        }
                        if(ae.getSource()==cancelButton)
                        {
                                setVisible(false);
                        }
                }
        }// END OF fontDialogBox CLASS

        //MAIN FUNCTION OF MYEDIT CLASS
        public static void main(String args[])
        {
                e= new myedit();
        }
}//END OF MYEDIT CLASS
```
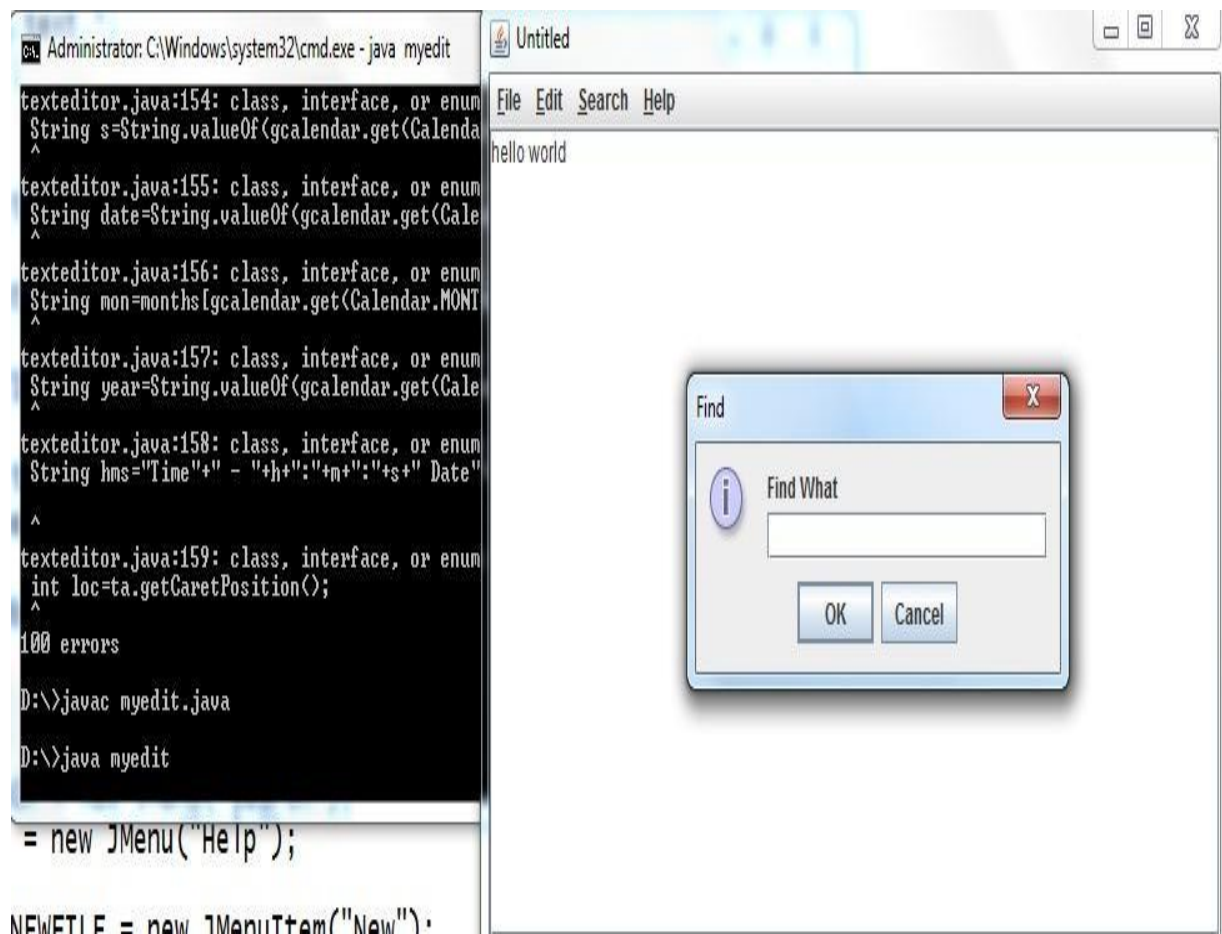
Output:

Result:

The Java Swing application for creating a Simple text editor was developed and tested successfully.