P E S COLLEGE OF ENGINEERING

(An Autonomous Institution Affiliated to VTU, Belagavi)

Mandya - 571401, Karnataka

Programming in Java Lab Manual

(P18ISL48)

Prepared by,

PUTTASWAMY B. S.

Assistant Professor

Information Science and Engineering

P E S College of Engineering

Mandya – 571401

(2021-22)

Vision

"PESCE shall be a leading institution imparting quality engineering and management education developing creative and socially responsible professionals."

"PESCE ಸೃಜನಶೀಲ ಮತ್ತು ಸಾಮಾಜಿಕ ಜವಾಬ್ದಾರಿಯುತ ವೃತ್ತಿಪರರನ್ನು ಅಭಿವೃದ್ಧಿಪಡಿಸುವ ಗುಣಮಟ್ಟದ ಎಂಜಿನಿಯರಿಂಗ್ ಮತ್ತು ನಿರ್ವಹಣಾ ಶಿಕ್ಷಣವನ್ನು ನೀಡುವ ಪ್ರಮುಖ ಸಂಸ್ಥೆಯಾಗಿದೆ."

Mission

- Provide state of the art infrastructure, motivate the faculty to be proficient in their field of specialization and adopt best teaching-learning practices.

  ಅತ್ಯಾಧುನಿಕ ಮೂಲಸೌಕರ್ಯಗಳನ್ನು ಒದಗಿಸಿ, ಬೋಧಕವರ್ಗವನ್ನು ತಮ್ಮ ವಿಶೇಷ ಕ್ಷೇತ್ರದಲ್ಲಿ ಪ್ರವೀಣರಾಗುವಂತೆ ಪ್ರೇರೇಪಿಸಿ ಮತ್ತು ಅತ್ಯುತ್ತಮ ಬೋಧನೆ-ಕಲಿಕೆಯ ಅಭ್ಯಾಸಗಳನ್ನು ಅಳವಡಿಸಿಕೊಳ್ಳಿ.

- Impart engineering and managerial skills through competent and committed faculty using outcome based educational curriculum.

  ಫಲಿತಾಂಶ ಆಧಾರಿತ ಶೈಕ್ಷಣಿಕ ಪಠ್ಯಕ್ರಮವನ್ನು ಬಳಸಿಕೊಂಡು ಸಮರ್ಥ ಮತ್ತು ಬದ್ಧ ಅಧ್ಯಾಪಕರ ಮೂಲಕ ಎಂಜಿನಿಯರಿಂಗ್ ಮತ್ತು ನಿರ್ವಾಹಕ ಕೌಶಲ್ಯಗಳನ್ನು ನೀಡಿ.

- Inculcate professional ethics, leadership qualities and entrepreneurial skills to meet the societal needs.

  ಸಾಮಾಜಿಕ ಅಗತ್ಯಗಳನ್ನು ಪೂರೈಸಲು ವೃತ್ತಿಪರ ನೈತಿಕತೆ, ನಾಯಕತ್ವ ಗುಣಗಳು ಮತ್ತು ಉದ್ಯಮಶೀಲತೆಯ ಕೌಶಲ್ಯಗಳನ್ನು ರೂಢಿಸಿಕೊಳ್ಳಿ.

- Promote research, product development and industry-institution interaction.

  ಸಂಶೋಧನೆ, ಉತ್ಪನ್ನ ಅಭಿವೃದ್ಧಿ ಮತ್ತು ಉದ್ಯಮ-ಸಂಸ್ಥೆಗಳ ಪರಸ್ಪರ ಕ್ರಿಯೆಯನ್ನು ಉತ್ತೇಜಿಸಿ.

About the Department

The Department of Information science and Engineering takes pride in producing quality engineers over the past 20 years. The credit for all the flowery results goes to the highly motivating staff, from whom all students draw inspiration. The Department was started in the year 2000. The present intake of the undergraduate program is 60. The department has well equipped classrooms, computer laboratories with high-end systems, department library and good collection of software's. Also a research centre is a major credential to our department. We are proud to produce the first PhD student in our college. Faculty members of the department are involved in research activities in different fields such as Medical Image Processing, Pattern Recognition, and Data Mining etc. The department is using Outcome-based education (OBE), which is a recurring education reform model, and it is affiliated to Visvesvaraya Technological University (VTU). The department has achieved good Placement, conducted International /national Conferences and other sponsored short-term courses, workshops, National seminars and symposia. The laboratory facilities and the Internet access are available round the clock to the staff and students of the Information Science and Engineering.

ಮಾಹಿತಿ ವಿಜ್ಞಾನ ಮತ್ತು ಎಂಜಿನಿಯರಿಂಗ್ ವಿಭಾಗವು ಕಳೆದ 20 ವರ್ಷಗಳಲ್ಲಿ ಗುಣಮಟ್ಟದ ಎಂಜಿನಿಯರ್‌ಗಳನ್ನು ಉತ್ಪಾದಿಸುವಲ್ಲಿ ಹೆಮ್ಮೆ ಪಡುತ್ತದೆ. ಎಲ್ಲಾ ಹೂವುಗಳ ಫಲಿತಾಂಶಗಳ ಕ್ರೆಡಿಟ್ ಹೆಚ್ಚು ಪ್ರೇರೇಪಿಸುವ ಸಿಬ್ಬಂದಿಗೆ ಸಲ್ಲುತ್ತದೆ, ಅವರಿಂದ ಎಲ್ಲಾ ವಿದ್ಯಾರ್ಥಿಗಳು ಸ್ಫೂರ್ತಿ ಪಡೆಯುತ್ತಾರೆ. ಇಲಾಖೆಯು 2000 ನೇ ವರ್ಷದಲ್ಲಿ ಆರಂಭವಾಯಿತು. ಪದವಿಪೂರ್ವ ಕಾರ್ಯಕ್ರಮದ ಪ್ರಸ್ತುತ ಸೇವನೆಯು 60. ಇಲಾಖೆಯು ಸುಸಜ್ಜಿತವಾದ ತರಗತಿ ಕೊಠಡಿಗಳು, ಉನ್ನತ ಮಟ್ಟದ ವ್ಯವಸ್ಥೆಗಳೊಂದಿಗೆ ಕಂಪ್ಯೂಟರ್ ಪ್ರಯೋಗಾಲಯಗಳು, ಇಲಾಖೆಯ ಗ್ರಂಥಾಲಯ ಮತ್ತು ಸಾಫ್ಟ್‌ವೇರ್‌ಗಳ ಉತ್ತಮ ಸಂಗ್ರಹವನ್ನು ಹೊಂದಿದೆ. ಅಲ್ಲದೆ ಒಂದು ಸಂಶೋಧನಾ ಕೇಂದ್ರವು ನಮ್ಮ ಇಲಾಖೆಗೆ ಪ್ರಮುಖ ರುಜುವಾತು. ನಮ್ಮ ಕಾಲೇಜಿನಲ್ಲಿ ಮೊದಲ ಪಿಎಚ್‌ಡಿ ವಿದ್ಯಾರ್ಥಿಯನ್ನು ತಯಾರಿಸಲು ನಮಗೆ ಹೆಮ್ಮೆ ಇದೆ. ಇಲಾಖೆಯ ಅಧ್ಯಾಪಕರು ವೈದ್ಯಕೀಯ ಚಿತ್ರ ಸಂಸ್ಕರಣೆ, ಪ್ಯಾಟರ್ನ್ ರೆಕಗ್ನಿಷನ್, ಮತ್ತು ಡೇಟಾ ಮೈನಿಂಗ್ ಮುಂತಾದ ವಿವಿಧ ಕ್ಷೇತ್ರಗಳಲ್ಲಿ ಸಂಶೋಧನಾ ಚಟುವಟಿಕೆಗಳಲ್ಲಿ ತೊಡಗಿಸಿಕೊಂಡಿದ್ದಾರೆ. ಇಲಾಖೆಯು ಫಲಿತಾಂಶ ಆಧಾರಿತ ಶಿಕ್ಷಣವನ್ನು (ಒಬಿಇ) ಬಳಸುತ್ತಿದೆ, ಇದು ಪುನರಾವರ್ತಿತ ಶಿಕ್ಷಣ ಸುಧಾರಣಾ ಮಾದರಿಯಾಗಿದೆ, ಮತ್ತು ಇದು ವಿಶ್ವೇಶ್ವರಯ್ಯ ತಾಂತ್ರಿಕ ವಿಶ್ವವಿದ್ಯಾಲಯಕ್ಕೆ (ವಿಟಿಯು) ಸಂಯೋಜಿತವಾಗಿದೆ. ಇಲಾಖೆಯು ಉತ್ತಮ ಉದ್ಯೋಗವನ್ನು ಸಾಧಿಸಿದೆ, ಅಂತರಾಷ್ಟ್ರೀಯ /ರಾಷ್ಟ್ರೀಯ ಸಮ್ಮೇಳನಗಳನ್ನು ಮತ್ತು ಇತರ ಪ್ರಾಯೋಜಿತ ಅಲ್ಪಾವಧಿ ಕೋರ್ಸ್‌ಗಳು, ಕಾರ್ಯಾಗಾರಗಳು, ರಾಷ್ಟ್ರೀಯ ವಿಚಾರಗೋಷ್ಠಿಗಳು ಮತ್ತು ವಿಚಾರ ಸಂಕಿರಣಗಳನ್ನು ನಡೆಸಿದೆ. ಪ್ರಯೋಗಾಲಯದ ಸೌಲಭ್ಯಗಳು ಮತ್ತು ಇಂಟರ್ನೆಟ್ ಪ್ರವೇಶವು ಸಿಬ್ಬಂದಿ ಮತ್ತು ಮಾಹಿತಿ ವಿಜ್ಞಾನ ಮತ್ತು ಎಂಜಿನಿಯರಿಂಗ್‌ನ ವಿದ್ಯಾರ್ಥಿಗಳಿಗೆ 24 ಗಂಟೆಯೂ ಲಭ್ಯವಿದೆ.

Vision

"The department strives to equip our graduates with Knowledge and Skills to contribute significantly to Information Science & Engineering and enhance quality research for the benefit of society".

"ಮಾಹಿತಿ ವಿಜ್ಞಾನ ಮತ್ತು ಎಂಜಿನಿಯರಿಂಗ್‌ಗೆ ಗಣನೀಯ ಕೊಡುಗೆ ನೀಡಲು ಮತ್ತು ಸಮಾಜದ ಪ್ರಯೋಜನಕ್ಕಾಗಿ ಗುಣಮಟ್ಟದ ಸಂಶೋಧನೆಯನ್ನು ಹೆಚ್ಚಿಸಲು ನಮ್ಮ ಪದವೀಧರರನ್ನು ಜ್ಞಾನ ಮತ್ತು ಕೌಶಲ್ಯಗಳೊಂದಿಗೆ ಸಜ್ಜುಗೊಳಿಸಲು ಇಲಾಖೆಯು ಶ್ರಮಿಸುತ್ತದೆ."

Mission

- To provide students with state of art facilities and tools of Information Science & Engineering to become productive, global citizens and life-long learners.
  ವಿದ್ಯಾರ್ಥಿಗಳಿಗೆ ಅತ್ಯಾಧುನಿಕ ಸೌಲಭ್ಯಗಳು ಮತ್ತು ಮಾಹಿತಿ ವಿಜ್ಞಾನ ಮತ್ತು ಎಂಜಿನಿಯರಿಂಗ್ ಉಪಕರಣಗಳನ್ನು ಉತ್ಪಾದಕ, ಜಾಗತಿಕ ನಾಗರಿಕರು ಮತ್ತು ಜೀವನಪರ್ಯಂತ ಕಲಿಯುವವರನ್ನಾಗಿ ಮಾಡಲು.
- To prepare students for careers in IT industry, Higher education and Research.
  ಐಟಿ ಉದ್ಯಮ, ಉನ್ನತ ಶಿಕ್ಷಣ ಮತ್ತು ಸಂಶೋಧನೆಗಾಗಿ ವಿದ್ಯಾರ್ಥಿಗಳನ್ನು ತಯಾರಿಸಲು.
- To inculcate leadership qualities among students to make them competent Information Science & Engineering professionals or entrepreneurs.
  ವಿದ್ಯಾರ್ಥಿಗಳಲ್ಲಿ ನಾಯಕತ್ವ ಗುಣಗಳನ್ನು ಬೆಳೆಸಲು ಅವರನ್ನು ಸಮರ್ಥ ಮಾಹಿತಿ ವಿಜ್ಞಾನ ಮತ್ತು ಎಂಜಿನಿಯರಿಂಗ್ ವೃತ್ತಿಪರರು ಅಥವಾ ಉದ್ಯಮಿಗಳನ್ನಾಗಿ ಮಾಡಲು.

Program Educational Objectives (PES's)

PEO1: Establish a productive Information Science & Engineering career in industry, government or academia.

PEO2: Interact with their peers in other disciplines by exhibiting professionalism and team work to contribute to the economic growth of the country.

PEO3: Promote the development of innovative systems and solutions to the problems in Information Science using hardware and software integration.

PEO4: Pursue higher studies in Engineering, Management or Research.

Program Educational Objectives (PES's)

PSO1. Analyze, design, develop and test the principles of System software and Database concepts for computer-based systems.

PSO2. Develop computer communication systems and applications for Information security.

PSO3. Apply the knowledge of Information Science and Engineering to solve any software and hardware related problems and to organize, manage and monitor IT Infrastructure.

Program Outcome's (PO's)

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

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

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

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

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

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

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

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

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

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

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

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

General Laboratory Instructions

1. Students are advised to come to the laboratory at least 5 minutes before (to the starting time), those who come after 5 minutes will not be allowed into the lab.
2. Plan your task properly much before to the commencement, come prepared to the lab with the synopsis / program / experiment details.
3. Student should enter into the laboratory with:

a. Laboratory observation notes with all the details (Problem statement, Aim, Algorithm, Procedure, Program, Expected Output, etc.,) filled in for the lab session.

b. Laboratory Record updated up to the last session experiments and other utensils (if any) needed in the lab.

c. Proper Dress code and Identity card.

4. Sign in the laboratory login register, write the TIME-IN, and occupy the computer system allotted to you by the faculty.
5. Execute your task in the laboratory, and record the results / output in the lab observation note book, and get certified by the concerned faculty.
6. All the students should be polite and cooperative with the laboratory staff, must maintain the discipline and decency in the laboratory.
7. Computer labs are established with sophisticated and high end branded systems, which should be utilized properly.
8. Students / Faculty must keep their mobile phones in SWITCHED OFF mode during the lab sessions. Misuse of the equipment, misbehaviors with the staff and systems etc., will attract severe punishment.
9. Students must take the permission of the faculty in case of any urgency to go out; if anybody found loitering outside the lab / class without permission during working hours will be treated seriously and punished appropriately.
10. Students should LOG OFF/ SHUT DOWN the computer system before he/she leaves the lab after completing the task (experiment) in all aspects. He/she must ensure the system / seat is kept properly.

Course Overview

A Java Programming lab manual is intended to provide a basic knowledge of object oriented programming concepts (objects, classes and subclasses, methods, interfaces, packages, multithreading etc,) and their expression in Java for students. It also provides practice using non-object aspects of Java (loops, conditionals, etc.). To develop software development skills in java programming and Students will have the proficiency to develop projects in java programming. The course helps the students to solve the inter disciplinary applications through java programming.

Course Objectives

The objectives of this course are to make students to learn,

1. Apply object oriented concepts to real world problems in Java
2. Apply the concepts of inheritance and interfaces in Java
3. Demonstrate usage of packages, string handling and exception handling in Java
4. Illustrate need for multi-threading, generics and file handling in Java

Course Outcomes

1. Implement Java programs using object oriented concepts
2. Demonstrate features of Java involving Interfaces, Packages, String handling and Exception Handling
3. Apply the concepts of multi-threading, generics and files in Java

P E S Education Trust(R), Mandya

P E S College of Engineering

(An Autonomous Institution Affiliated to VTU, Belagavi)

Department of Information Science & Engineering

SYLLABUS

Subject Name: Programming in Java Laboratory						
Subject Code: P18ISL48				Category : Core Lab		
Hours / Week			Credits	Maximum Marks		
L	T	P	C	CIE	SEE	Total
-	-	3	1.5	50	50	100

List of Experiments

1 a.	Write a java program to check whether a given number is "prime number" and "palindrome number or not"
b.	Write a java program to design a class to represent a bank account and include the following members. Date members are: Name of depositor, Account Number, Type of account and Balance amount in the account. Methods are: To assign initial values, to deposit an account, to withdraw an amount after checking balance and to display the name and balance.
2 a.	Write a Java program to sort an array of positive integers of a given array, in the sorted array the value of the first element should be maximum, second value should be minimum value, third should be second maximum, fourth second be second minimum and so on.
b.	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:

<table>
<tr><td></td><td>

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

</td></tr>
<tr><td>3 a.</td><td>Write a program to read an array of n elements and implement following operations using Arrays class methods.
i) Perform Binary Search by reading element to be searched
ii) Initialize two arrays and check if the arrays are equal or not.</td></tr>
<tr><td>b.</td><td>A sales person is received commission based on the sales he makes as shown by the following table:
<table>
<tr><th>Sale</th><th>Commission</th></tr>
<tr><td>Up to Rs. 50,000</td><td>5% on sales value</td></tr>
<tr><td>More than Rs. 50,000 and up to Rs. 80,000</td><td>8% on sales value</td></tr>
<tr><td>More than Rs. 80,000 and up to Rs. 1,00,000</td><td>10% on sales value</td></tr>
<tr><td>More than Rs. 1,00,000</td><td>12% on sales value</td></tr>
</table>
Write a java program to define a class commission contains employee name, employee number and sales data member and methods input(), compute() and display() methods. Write a main method to create object of a class and call the above member methods.</td></tr>
<tr><td>4 a.</td><td>Write a Java program to separate even and odd numbers of a given array of integers. Put all even numbers first, and then odd numbers. Also find the number of even and odd integers in a given array of integers. Array elements are user Inputs.</td></tr>
<tr><td>b.</td><td>Write a class Worker and derive class DailyWorker and SalariedWorker from it. Every worker has a name and salary_rate. Write a method comPay(int hours) to compute the week pay of every worker. A daily worker is paid on the basis of number of days he work. The SalariedWorker gets paid the wage for 40hours a week no matter what</td></tr>
</table>

	actual hours is. Test this program to calculate the pay of workers. Write the program using the concept of polymorphism.
5 a.	Write a java program to determine the addition of two matrices.
b.	Write a program to create an abstract class shape with two instance variables d(length of the side of polygon)and sides(no. of sides of a polygon),constructor and abstract method area. Create subclass polygon and circle. For polygon class create object references for triangle, square and hexagon. Invoke method area for objects of polygon and circle classes. Write constructors for subclasses.
6 a.	Write a java program to sort an array of strings in alphabetical order using array.sort() method.
b.	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.
7 a.	Java Program to Find Largest Between Three Numbers Using Ternary Operator
b.	Write a java application to generate pay slip for different category of employees using the concept of inheritance
8 a.	Write a Java program to display the Fibonacci series
b.	Write a java program for array implementation of stack using the concept of interface and exception handling
9 a.	Write a program to read a string. Replace a character at the specified index in the String in Java
b.	Write a JAVA program that creates threads by extending Thread class First thread display “Good Morning “every 1 sec, the second thread displays “Hello “every 2 seconds and the third display “Welcome” every 3 seconds
10 a.	Write a Java Program to find all substrings of a given string
b.	Write a Java program to implement currency converter, distance converter and time converter using packages

INTRODUCTION TO JAVA PROGRAMMING

What is Java Programming?

Java is a general-purpose computer programming language that is concurrent, class-based, object-oriented, and specifically designed to have as few (lesser) implementation dependencies as possible. It is a computing platform for application development. Java is fast, secure, and reliable, therefore. It is widely used for developing Java applications in laptops, data centers, game consoles, scientific supercomputers, cell phones, etc.

- **Platform independent** - We can write Java code in one platform (operating system) and run on another platform without any modification.
- **Object-oriented** - Java is an object-oriented language. This helps to make our Java code more flexible and reusable.
- **Speed** - Well optimized Java code is nearly as fast as lower-level languages like C++ and much faster than Python, PHP, etc.

Why Learn Java?

Java is a platform-independent language. We can write Java code in one platform and run it in another platform ((Windows, Mac, Linux). Java is a general-purpose, easy to learn, simple to use, open source, free, secure, fast, powerful and huge community support programming language with a wide range of applications. It's used for developing mobile and desktop applications, big data processing, embedded systems, and so on. Java is an object-oriented programming language. It helps in code reusability. Also Java is close to C++ and C#, it makes it easy for programmers to switch to Java or vice versa.

History:

Java's history is very interesting. It is a programming language created in 1991. James Gosling, Mike Sheridan, and Patrick Naughton, a

team of Sun engineers known as the **Green team** initiated the Java language in 1991. **Sun Microsystems** released its first public implementation in 1996 as **Java 1.0**. It provides no-cost -run-times on popular platforms. Java1.0 compiler was re-written in Java by Arthur Van Hoff to strictly comply with its specifications. With the arrival of Java 2, new versions had multiple configurations built for different types of platforms.

In 1997, Sun Microsystems approached the ISO standards body and later formalized Java, but it soon withdrew from the process. At one time, Sun made most of its Java implementations available without charge, despite their proprietary software status. Sun generated revenue from Java through the selling of licenses for specialized products such as the Java Enterprise System.

On November 13, 2006, Sun released much of its Java virtual machine as free, open-source software. On May 8, 2007, Sun finished the process, making all of its JVM's core code available under open-source distribution terms.

The principles for creating java were simple, robust, secured, high performance, portable, multi-threaded, interpreted, dynamic, etc. In 1995 Java was developed by **James Gosling**, who is known as the Father of Java. Currently, Java is used in mobile devices, internet programming, games, e-business, etc.

Java programming language is named JAVA. Why?

After the name OAK, the team decided to give a new name to it and the suggested words were Silk, Jolt, revolutionary, DNA, dynamic, etc. These all names were easy to spell and fun to say, but they all wanted the name to reflect the essence of technology. In accordance with James Gosling, **Java** the among the top names along with **Silk**, and since java was a unique name so most of them preferred it.

Java is the name of an **island** in Indonesia where the first coffee(named java coffee) was produced. And this name was chosen by James Gosling while having coffee near his office. Note that Java is just a name, not an acronym.

Java Terminology:

Before learning Java, one must be familiar with these common terms of Java.

1. Java Virtual Machine(JVM): This is generally referred to as JVM. There are three execution phases of a program. They are written, compile and run the program.

- Writing a program is done by a java programmer like you and me.
- The compilation is done by the **JAVAC** compiler which is a primary Java compiler included in the Java development kit (JDK). It takes the Java program as input and generates byte-code as output.
- In the Running phase of a program, **JVM** executes the byte-code generated by the compiler.

Now, we understood that the function of Java Virtual Machine is to execute the byte-code produced by the compiler. Every Operating System has a different JVM but the output they produce after the execution of bytecode is the same across all the operating systems. This is why Java is known as a **platform-independent language.**

2. Byte-code in the **Development process:** As discussed, the Javac compiler of JDK compiles the java source code into byte-code so that it can be executed by JVM. It is saved as **.class** file by the compiler. To view the byte-code, a disassembler like javap can be used.

3. Java Development Kit(JDK): While we were using the term JDK when we learn about byte-code and JVM. So, as the name suggests, it is a complete Java development kit that includes everything including compiler, Java Runtime Environment (JRE), java debuggers, java docs, etc. For the program to execute in java, we need to install JDK on our computer in order to create, compile and run the java program.

4. Java Runtime Environment (JRE): JDK includes JRE. JRE installation on our computers allows the java program to run, however, we cannot compile it. JRE includes a browser, JVM, applet supports, and plugins. For running the java program, a computer needs JRE.

5. Garbage Collector: In Java, programmers can't delete the objects. To delete or recollect that memory JVM has a program called Garbage Collector.

Garbage Collectors can recollect the objects that are not referenced. So Java makes the life of a programmer easy by handling memory management. However, programmers should be careful about their code whether they are using objects that have been used for a long time. Because Garbage cannot recover the memory of objects being referenced.

6. ClassPath: The classpath is the file path where the java runtime and Java compiler look for **.class** files to load. By default, JDK provides many libraries. If you want to include external libraries they should be added to the classpath.

Primary/Main Features of Java

1. Platform Independent: Compiler converts source code to bytecode and then the JVM executes the bytecode generated by the compiler. This bytecode can run on any platform be it Windows, Linux, macOS which means if we compile a program on Windows, then we can run it on Linux and vice versa. Each operating system has a different JVM, but the output produced by all the OS is the same after the execution of bytecode. That is why we call java a platform-independent language.

2. Object-Oriented Programming Language: Organizing the program in the terms of collection of objects is a way of object-oriented programming, each of which represents an instance of the class.

The four main concepts of Object-Oriented programming are:

- Abstraction
- Encapsulation
- Inheritance
- Polymorphism

3. Simple: Java is one of the simple languages as it does not have complex features like pointers, operator overloading, multiple inheritances, Explicit memory allocation.

4. Robust: Java language is robust which means reliable. It is developed in such a way that it puts a lot of effort into checking errors as early as possible, that is why the java compiler is able to detect even those errors that are not easy to detect by another programming language. The main features of java

that make it robust are garbage collection, Exception Handling, and memory allocation.

5. Secure: In java, we don't have pointers, so we cannot access out-of-bound arrays i.e it shows **ArrayIndexOutOfBound Exception** if we try to do so. That's why several security flaws like stack corruption or buffer overflow are impossible to exploit in Java.

6. Distributed: We can create distributed applications using the java programming language. Remote Method Invocation and Enterprise Java Beans are used for creating distributed applications in java. The java programs can be easily distributed on one or more systems that are connected to each other through an internet connection.

7. Multithreading: Java supports multithreading. It is a Java feature that allows concurrent execution of two or more parts of a program for maximum utilization of CPU.

8. Portable: As we know, java code written on one machine can be run on another machine. The platform-independent feature of java in which its platform-independent bytecode can be taken to any platform for execution makes java portable.

9. High Performance: Java architecture is defined in such a way that it reduces overhead during the runtime and at some time java uses Just In Time (JIT) compiler where the compiler compiles code on-demand basics where it only compiles those methods that are called making applications to execute faster.

10. Dynamic flexibility: Java being completely object-oriented gives us the flexibility to add classes, new methods to existing classes and even create new classes through sub-classes. Java even supports functions written in other languages such as C, C++ which are referred to as native methods.

11. Sandbox Execution: Java programs run in a separate space that allows user to execute their applications without affecting the underlying system with help of a bytecode verifier. Bytecode verifier also provides additional security as its role is to check the code for any violation of access.

12. Write Once Run Anywhere: As discussed above java application generates a '.class' file which corresponds to our applications(program) but

contains code in binary format. It provides ease t architecture-neutral ease as bytecode is not dependent on any machine architecture. It is the primary reason java is used in the enterprising IT industry globally worldwide.

13. Power of compilation and interpretation: Most languages are designed with purpose either they are compiled language or they are interpreted language. But java integrates arising enormous power as Java compiler compiles the source code to bytecode and JVM executes this bytecode to machine OS-dependent executable code.

Java Platforms / Editions:

There are 4 platforms or editions of Java:

1) Java SE (Java Standard Edition): It is a Java programming platform. It includes Java programming APIs such as java.lang, java.io, java.net, java.util, java.sql, java.math etc. It includes core topics like OOPs, String, Regex, Exception, Inner classes, Multithreading, I/O Stream, Networking, AWT, Swing, Reflection, Collection, etc.

2) Java EE (Java Enterprise Edition): It is an enterprise platform that is mainly used to develop web and enterprise applications. It is built on top of the Java SE platform. It includes topics like Servlet, JSP, Web Services, EJB, JPA, etc.

3**) Java ME (Java Micro Edition):** It is a micro platform that is dedicated to mobile applications.

4) JavaFX: It is used to develop rich internet applications. It uses a lightweight user interface API.

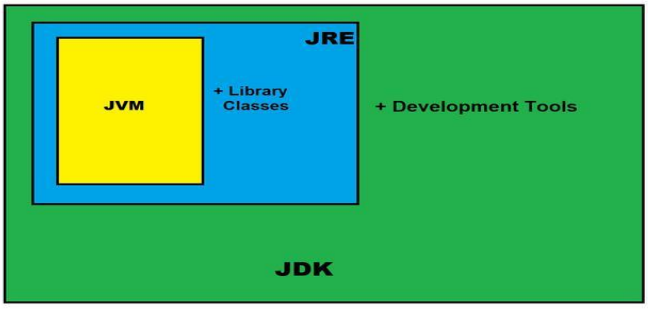
JDK (Java Development Kit):

The Java Development Kit (JDK) is a cross-plat formed software development environment that offers a collection of tools and libraries necessary for developing Java-based software applications and applets. It is a core package used in Java, along with the **JVM (Java Virtual Machine)** and the JRE (Java Runtime Environment).

Beginners often get confused with JRE and JDK, if you are only interested in running Java programs on your machine then you can easily do it using Java Runtime Environment. However, if you would like to develop a Java-based

software application then along with JRE you may need some additional necessary tools, which is called JDK.

JDK=JRE+Development Tools



The Java Development Kit is an implementation of one of the Java Platform:

- Standard Edition (Java SE),
- Java Enterprise Edition (Java EE),
- Micro Edition (Java ME),

Contents of JDK:

The JDK has a private Java Virtual Machine (JVM) and a few other resources necessary for the development of a Java Application.

JDK contains:

- Java Runtime Environment (JRE),
- An interpreter/loader (Java),
- A compiler (javac),
- An archiver (jar) and many more.

The Java Runtime Environment in JDK is usually called Private Runtime because it is separated from the regular JRE and has extra contents. The Private Runtime in JDK contains a JVM and all the class libraries present in the production environment, as well as additional libraries useful to developers, e.g, internationalization libraries and the IDL libraries.

Most Popular JDKs:

- **Oracle JDK:** the most popular JDK and the main distributor of Java11,
- **OpenJDK:** Ready for use: JDK 15, JDK 14, and JMC,
- **Azul Systems Zing:** efficient and low latency JDK for Linux os,
- **Azul Systems:** based Zulu brand for Linux, Windows, Mac OS X,

- **IBM J9 JDK:** for AIX, Linux, Windows, and many other OS,
- **Amazon Corretto:** The newest options with the no-cost build of OpenJDK and long-term support.

JDK versions:

Version	Release date	End of Free Public Updates[1][4][5]	Extended Support Until
JDK Beta	1995	?	?
JDK 1.0	January 1996	?	?
JDK 1.1	February 1997	?	?
J2SE 1.2	December 1998	?	?
J2SE 1.3	May 2000	?	?
J2SE 1.4	February 2002	October 2008	February 2013
J2SE 5.0	September 2004	November 2009	April 2015
Java SE 6	December 2006	April 2013	December 2018
Java SE 7	July 2011	April 2015	July 2022
Java SE 8 (LTS)	March 2014	**January 2019 for Oracle (commercial)** Indefinitely for Oracle (personal use) At least May 2026 for AdoptOpenJDK At least May 2026 for Amazon Corretto	December 2030
Java SE 9	September 2017	March 2018 for OpenJDK	N/A
Java SE 10	March 2018	September 2018 for OpenJDK	N/A
Java SE 11 (LTS)	September 2018	At least October 2024 for AdoptOpenJDK At least September 2027 for Amazon Corretto	September 2026
Java SE 12	March 2019	September 2019 for OpenJDK	N/A
Java SE 13	September 2019	March 2020 for OpenJDK	N/A
Java SE 14	March 2020	September 2020 for OpenJDK	N/A
Java SE 15	September 2020	March 2021 for OpenJDK	N/A
Java SE 16	March 2021	September 2021 for OpenJDK	N/A
Java SE 17 (LTS)	September 2021	TBA	TBA
Legend: Old version	Older version, still maintained	**Latest version**	Future release		

Types of Java Applications:

There are mainly 4 types of applications that can be created using Java programming:

1) Standalone Application:

Standalone applications are also known as desktop applications or window-based applications. These are traditional software that we need to install on every machine. Examples of standalone application are Media player, antivirus, etc. AWT and Swing are used in Java for creating standalone applications.

2) Web Application:

An application that runs on the server side and creates a dynamic page is called a web application. Currently Servelet, JSP, Struts, Spring, Hibernate JSF etc. technologies are used for creating web applications in Java.

3) Enterprise Application:

An application that is distributed in nature, such as banking applications, etc. is called an enterprise application. It has advantages like high-level security, load balancing, and clustering. In Java, EJB is used for creating enterprise applications.

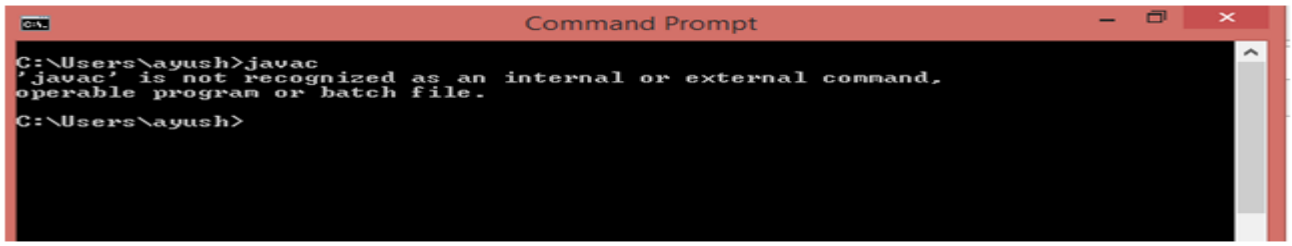
4) Mobile Application:

An application which is created for mobile devices is called a mobile application. Currently, Android and Java ME are used for creating mobile applications.

JDK installation steps:

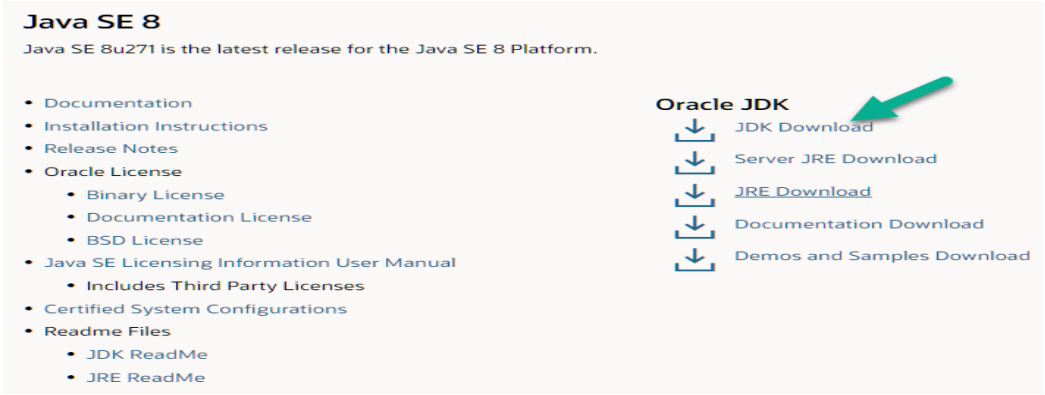
Step 1: Verify that it is already installed or not:

Check whether Java is already installed on the system or not. In my case, it is not installed therefore I need to install JDK 1.8 on my computer.



Step 2: Download JDK from the Site:

Go to link. Click on JDK Download for Java download JDK 8.

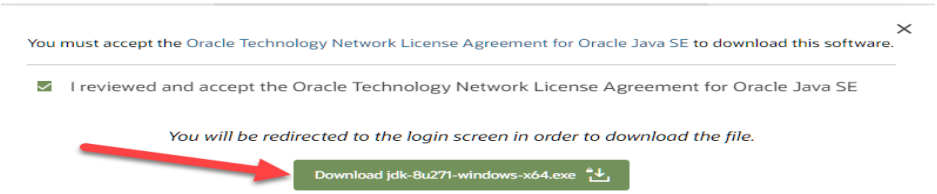


Step 2.1 Next,

1. Accept License Agreement
2. Download Java 8 JDK for your version 32 bit or JDK download 64 bit.

Solaris SPARC 64-bit	88.75 MB	jdk-8u271-solaris-sparcv9.tar.gz
Solaris x64 (SVR4 package)	134.42 MB	jdk-8u271-solaris-x64.tar.Z
Solaris x64	92.52 MB	jdk-8u271-solaris-x64.tar.gz
Windows x86	154.48 MB	jdk-8u271-windows-i586.exe
Windows x64	166.79 MB	jdk-8u271-windows-x64.exe

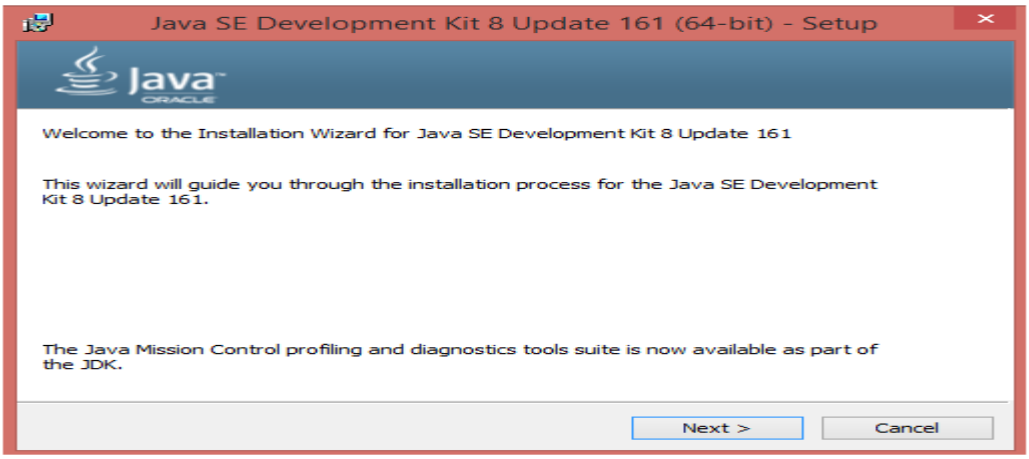
Step 2.2 When you click on the Installation link the popup will be open. Click on I reviewed and accept the Oracle Technology Network License Agreement for Oracle Java SE development kit and you will be redirected to the login page. If you don't have an oracle account you can easily sign up by adding basics details of yours.



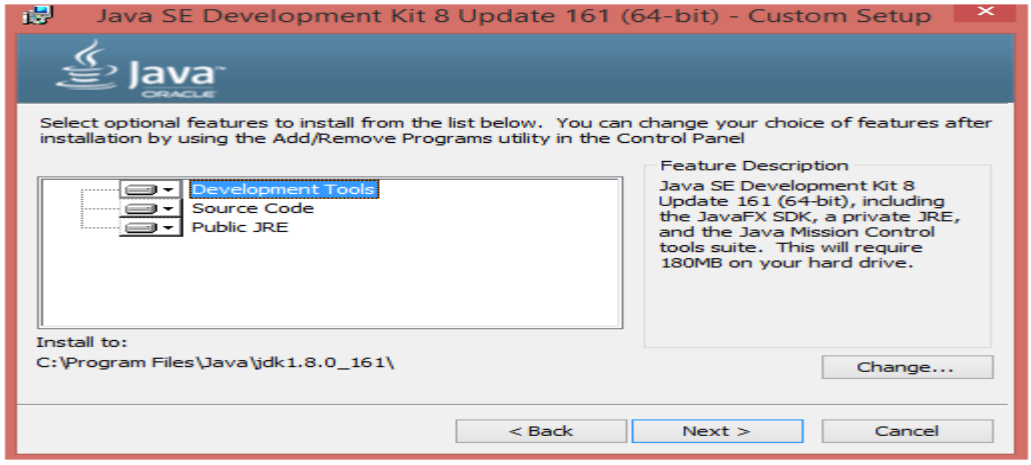
NOTE: You will be required to create an Oracle Account to start Java 8 download of the file.

Step 3: Install JDK:

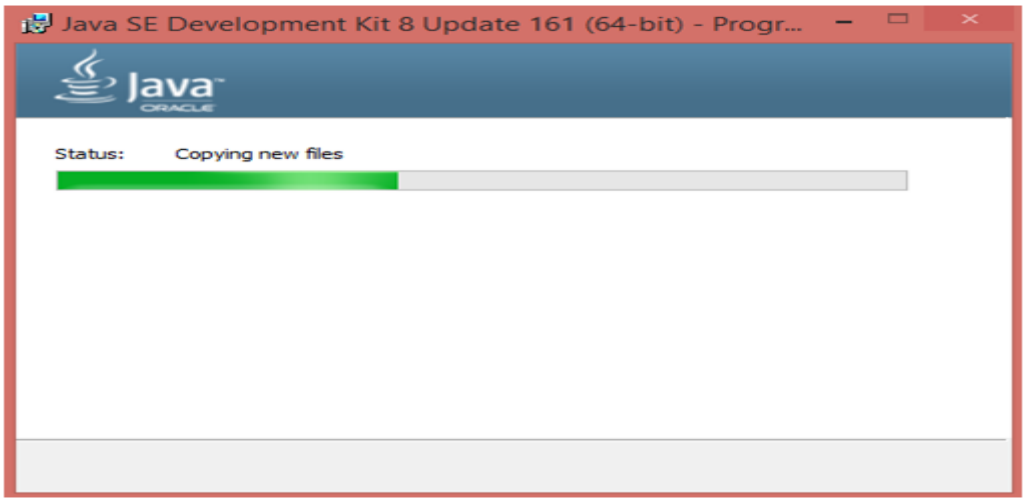
Once the Java JDK 8 download is complete, run the exe for install JDK. Click Next



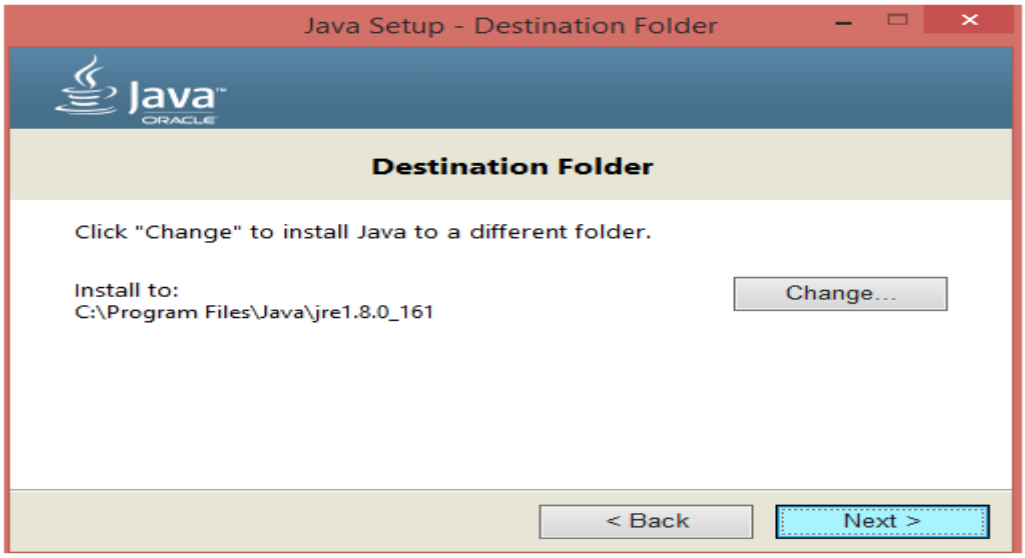
Click **Next** to continue



Just Choose Development Tools and click Next.

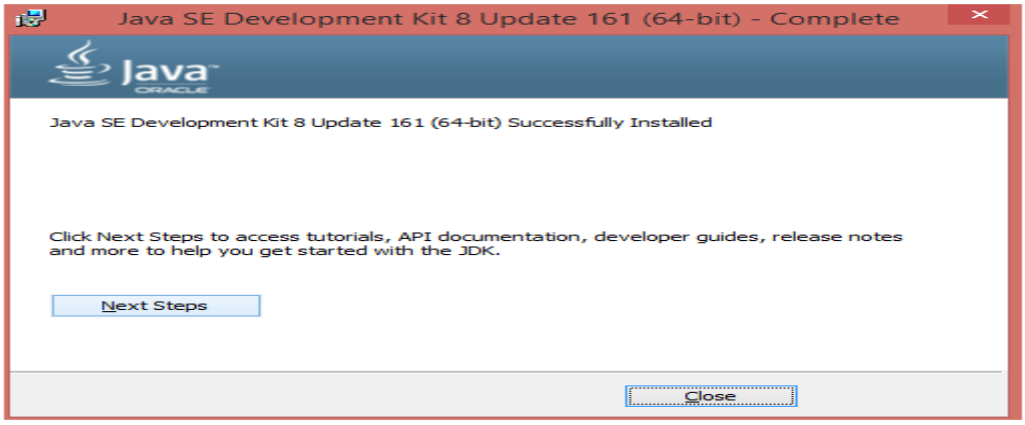


Set up is being ready.



Choose the Destination folder in which you want to install JDK. Click Next to continue with the installation.

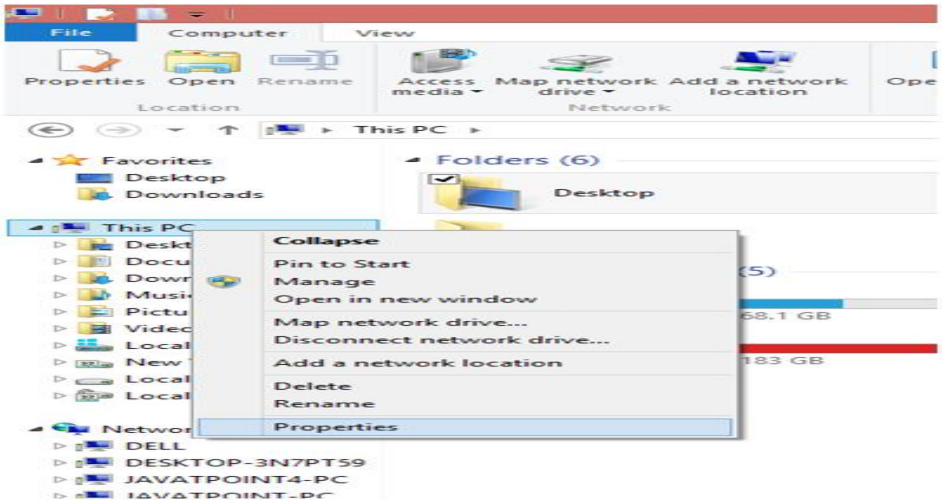
Set up is installing Java to the computer.



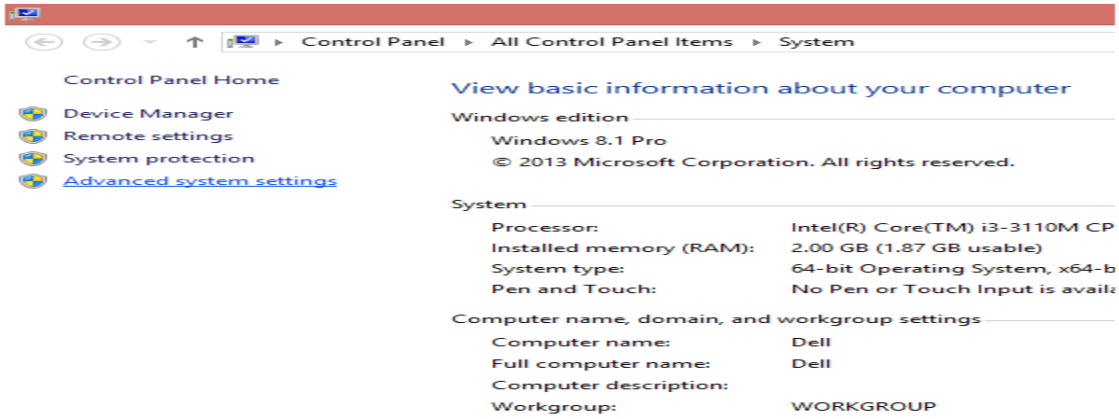
We have successfully installed Java SE development kit 8. Close the installation set up.

Step 4 : Set the Permanent Path

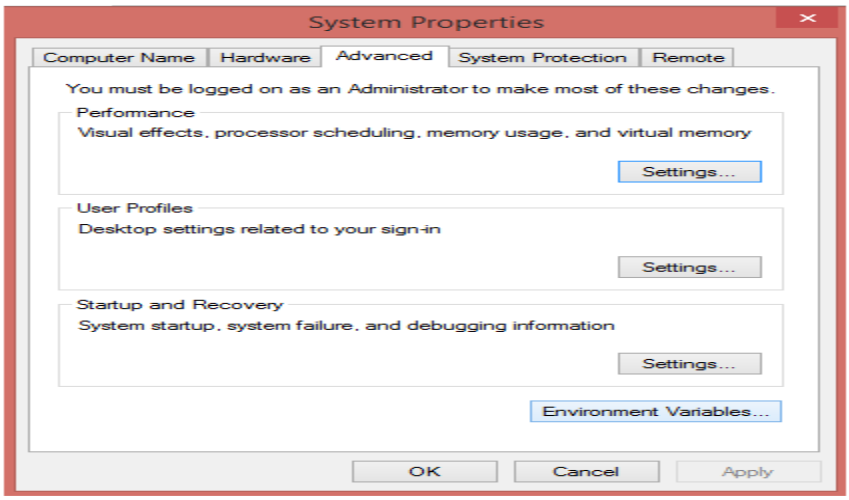
To execute Java applications from command line, we need to set Java Path. To set the path, follow the following steps.



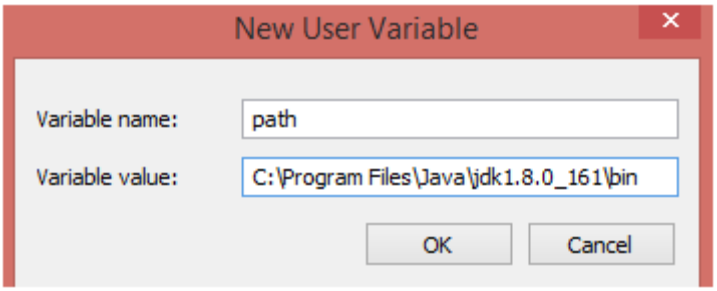
Right click on "this PC". It can be named as "My Computer" in some systems. Choose "properties" from the options.



The screen look alike the above image will open. Click on "Advanced system settings" to continue.



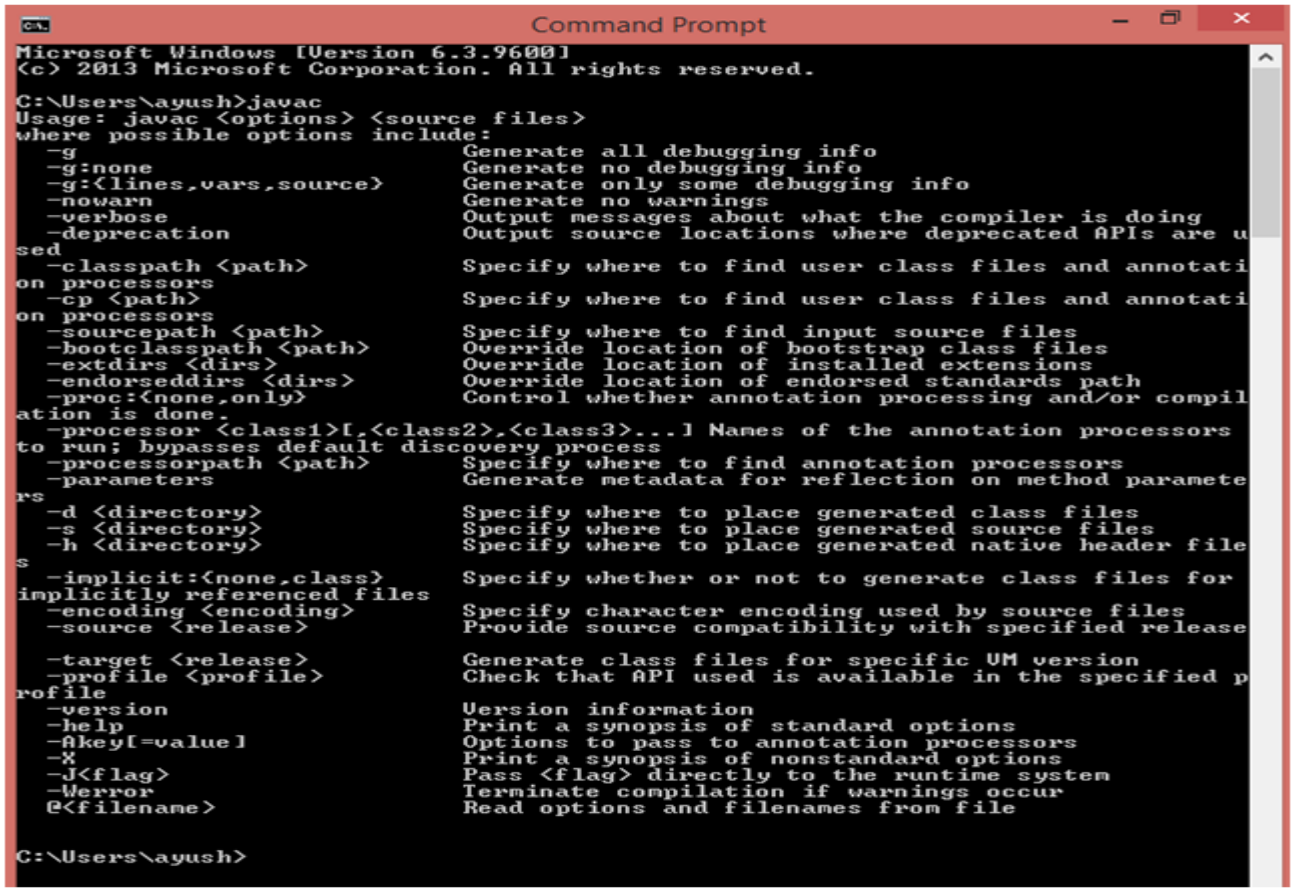
Above window will open. Click on "Environment Variables" to continue.



Enter "path" in variable name and enter the path to the bin folder inside your JDK in the variable value. Click OK.

Now Java Path has been set up. Open the Command prompt and type "**javac**" In case you have already open up the command prompt, I suggest you to close the existing window and reopen it again.

We will get javac executed as shown in the image below.



The Java has been installed on our system. Now, we need to configure IDEs like NetBeans or Eclipse in order to execute JavaFX applications.

Structure of Java Program:

```
// Demo Java program
// Importing classes from packages
import java.io.*;
  // Main class
public class ise
   {
      // Main driver method
    public static void main(String[] args)
    {
          // Print statement
        System.out.println("Welcome to ISE PESCE");
    }
}
```

Output

Welcone to ISEPESCE

Explanation:

1. **Comments:** Comments are used for explaining code and are used in a similar manner in Java or C or C++. Compilers ignore the comment entries and do not execute them. Comments can be of a single line or multiple lines.

Single line Comments:

Syntax:

// Single line comment

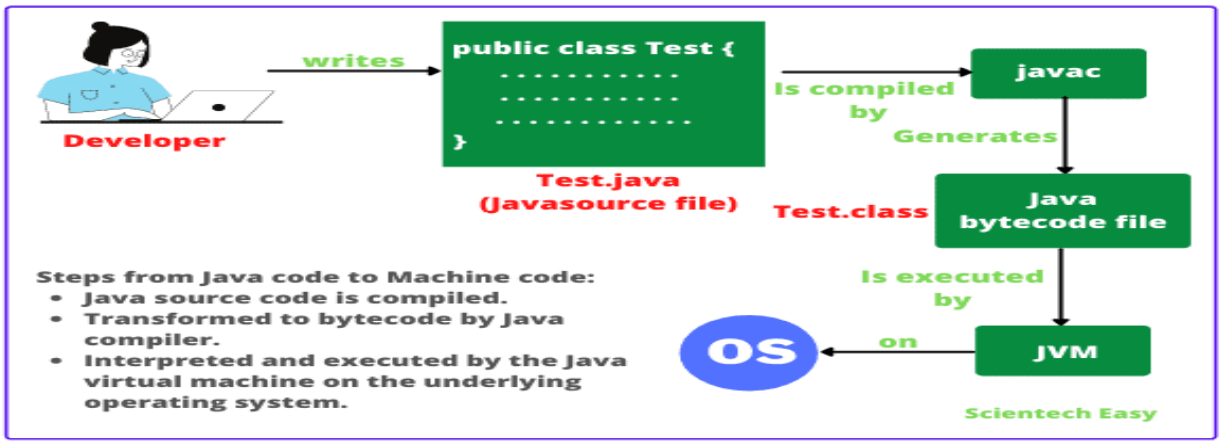
Multi-line comments:

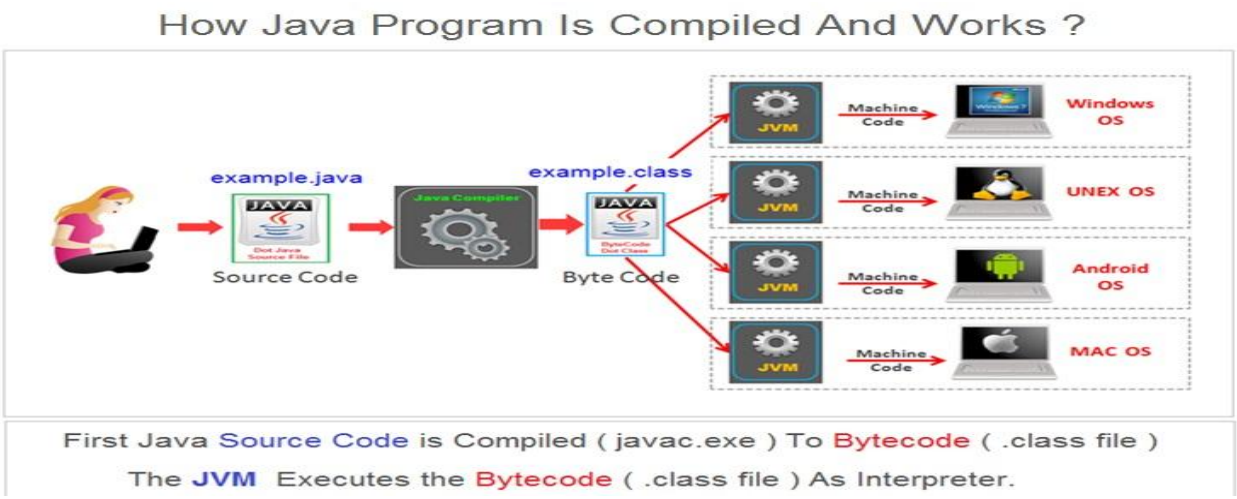
Syntax:

/* Multi line comments*/

2. **import java.io.*:** This means all the classes of io package can be imported. Java io package provides a set of input and output streams for reading and writing data to files or other input or output sources.
3. **class:** The class contains the data and methods to be used in the program. Methods define the behavior of the class. Class **GFG** has only one method Main in JAVA.
4. **static void Main(): static** keyword tells us that this method is accessible without instantiating the class.
5. **void:** keywords tell that this method will not return anything. The main**()** method is the entry point of our application.
6. **System.in:** This is the **standard input stream** that is used to read characters from the keyboard or any other standard input device.
7. **System.out:** This is the **standard output stream** that is used to produce the result of a program on an output device like the computer screen.
8. **println():** This method in Java is also used to display text on the console. It prints the text on the console and the cursor moves to the start of the next line at the console. The next printing takes place from the next line.

Everything in java, is represented in Class as an object including the main function.





Introduction to OOPs Concepts

Object-Oriented Programming System (OOPs) is a programming concept that works on the principles of abstraction, encapsulation, inheritance, and polymorphism. It allows users to create objects they want and create methods to handle those objects. The basic concept of OOPs is to create objects, re-use them throughout the program, and manipulate these objects to get results.

The main aim of OOP is to bind together the data and the functions that operate on them so that no other part of the code can access this data except that function.

Let us do discuss pre-requisites by polishing concepts of methods declaration and passing. Starting off with the method declaration, it consists of six components:

- **Access Modifier**: Defines **access type** of the method i.e. from where it can be accessed in your application. In Java, there 4 type of the access specifiers.
    - **public:** accessible in all class in your application.
    - **protected:** accessible within the package in which it is defined and in its **subclass(es)(including subclasses declared outside the package)**
    - **private:** accessible only within the class in which it is defined.
    - **default (declared/defined without using any modifier):** accessible within same class and package within which its class is defined.
- **The return type**: The data type of the value returned by the method or void if does not return a value.
- **Method Name**: the rules for field names apply to method names as well, but the convention is a little different.
- **Parameter list**: Comma separated list of the input parameters are defined, preceded with their data type, within the enclosed parenthesis. If there are no parameters, you must use empty parentheses ().
- **Exception list**: The exceptions you expect by the method can throw, you can specify these exception(s).
- **Method body**: it is enclosed between braces. The code you need to be executed to perform your intended operations.

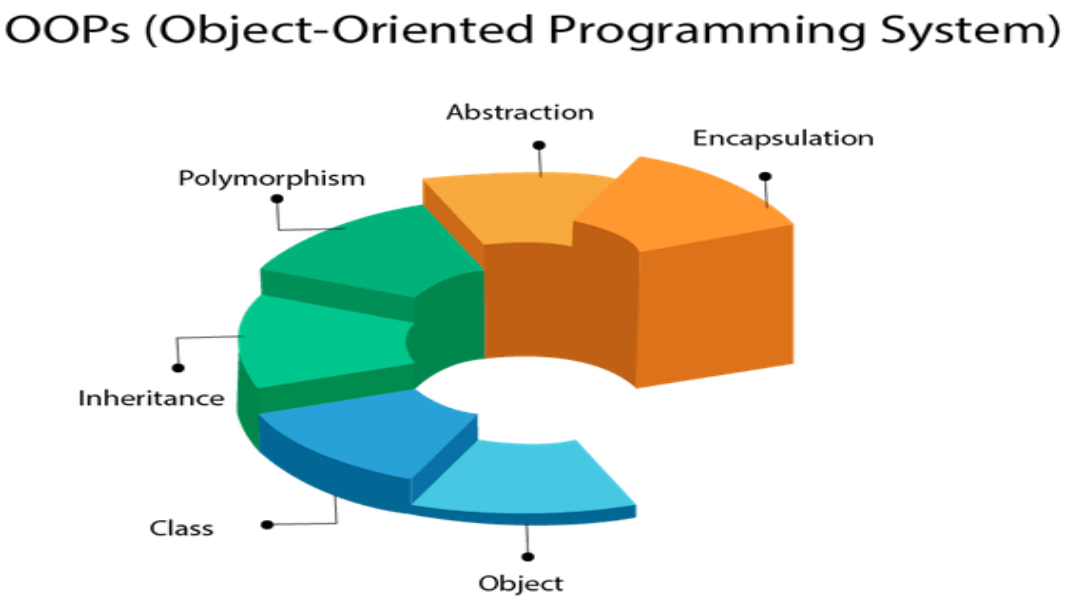
Message Passing: Objects communicate with one another by sending and receiving information to each other. A message for an object is a request for execution of a procedure and therefore will invoke a function in the receiving object that generates the desired results. Message passing involves specifying the name of the object, the name of the function and the information to be sent.

Now with basic prerequisite to step learning 4 pillar of OOPS is as follows. Let us start with learning about the different characteristics of an Object-Oriented Programming language.

OOPs Concepts are as follows:

1. Class

2. Object
3. Method and method passing
4. Pillars of OOPS
    - Abstraction
    - Encapsulation
    - Inheritance
    - Polymorphism
        - Compile-time polymorphism
        - Run-time polymorphism



Class:

A class is a user defined blueprint or prototype from which objects are created. It represents the set of properties or methods that are common to all objects of one type. In general, class declarations can include these components, in order:

1. **Modifiers**: A class can be public or has default access (Refer this for details).
2. **Class name:** The name should begin with a initial letter (capitalized by convention).
3. **Superclass(if any):** The name of the class's parent (superclass), if any, preceded by the keyword extends. A class can only extend (subclass) one parent.

4. **Interfaces(if any):** A comma-separated list of interfaces implemented by the class, if any, preceded by the keyword implements. A class can implement more than one interface.
5. **Body:** The class body surrounded by braces, {}.

```
class classname {
type instance variable 1;
type instance variable 2;
.
type instance variable n;
type methodname 1 (parameter list) {
// body od method
}
type methodname 2 (parameter list) {
// body od method
}
type methodnamen (parameter list) {
// body od method
}
 }
```

Object:

Object is a basic unit of Object Oriented Programming and represents the real life entities. A typical Java program creates many objects, which as you know, interact by invoking methods. An object consists of:

1. **State** : It is represented by attributes of an object. It also reflects the properties of an object.
2. **Behavior** : It is represented by methods of an object. It also reflects the response of an object with other objects.
3. **Identity** : It gives a unique name to an object and enables one object to interact with other objects.
4. **Method:** A method is a collection of statements that perform some specific task and return result to the caller. A method can perform some specific task without returning anything. Methods allow us to **reuse** the code without retyping the code. In Java, every method must be part of some

class which is different from languages like C, C++ and Python. Methods are **time savers** and help us to **reuse** the code without retyping the code.

Pillar 1: Data Abstraction:

Data Abstraction is the property by virtue of which only the essential details are displayed to the user. The trivial or the non-essentials units are not displayed to the user. Ex: A car is viewed as a car rather than its individual components.

Data Abstraction may also be defined as the process of identifying only the required characteristics of an object ignoring the irrelevant details. The properties and behaviors of an object differentiate it from other objects of similar type and also help in classifying/grouping the objects. Consider a real-life example of a man driving a car. The man only knows that pressing the accelerators will increase the speed of car or applying brakes will stop the car but he does not know about how on pressing the accelerator the speed is actually increasing, he does not know about the inner mechanism of the car or the implementation of accelerator, brakes etc in the car. This is what abstraction is.

In java, abstraction is achieved by interfaces and abstract classes. We can achieve 100% abstraction using interfaces.

Pillar 2: Encapsulation:

It is defined as the wrapping up of data under a single unit. It is the mechanism that binds together code and the data it manipulates. Another way to think about encapsulation is, it is a protective shield that prevents the data from being accessed by the code outside this shield.

- Technically in encapsulation, the variables or data of a class is hidden from any other class and can be accessed only through any member function of own class in which they are declared.

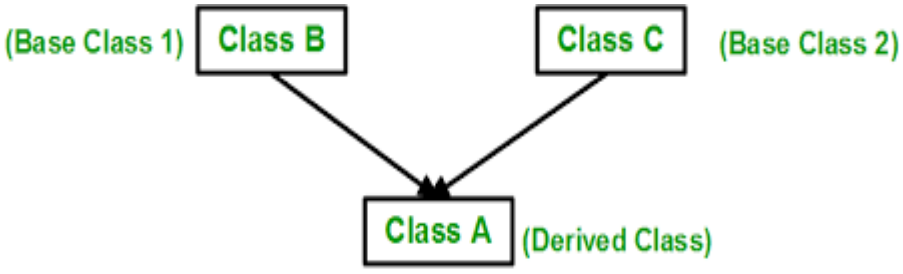
- As in encapsulation, the data in a class is hidden from other classes, so it is also known as **data-hiding**.
- Encapsulation can be achieved by Declaring all the variables in the class as private and writing public methods in the class to set and get the values of variables.

Pillar 3: Inheritence:

Inheritance is an important pillar of OOP(Object Oriented Programming). It is the mechanism in java by which one class is allow to inherit the features (fields and methods) of another class.

Let us discuss some of frequent used important terminologies:

- **Super Class:** The class whose features are inherited is known as superclass(or a base class or a parent class).
- **Sub Class:** The class that inherits the other class is known as subclass(or a derived class, extended class, or child class). The subclass can add its own fields and methods in addition to the superclass fields and methods.
- **Reusability:** Inheritance supports the concept of "reusability", i.e. when we want to create a new class and there is already a class that includes some of the code that we want, we can derive our new class from the existing class. By doing this, we are reusing the fields and methods of the existing class.



Pillar 4: Polymorphism:

It refers to the ability of OOPs programming languages to differentiate between entities with the same name efficiently. This is done by Java with the help of the signature and declaration of these entities.

Polymorphism in Java are mainly of 2 types:

1. Run time polymorphism
2. Compile time polymorphism

Advantage of OOPs over Procedure-oriented programming language

1) OOPs make development and maintenance easier, whereas, in a procedure-oriented programming language, it is not easy to manage if code grows as project size increases.

2) OOPs provide data hiding, whereas, in a procedure-oriented programming language, global data can be accessed from anywhere.

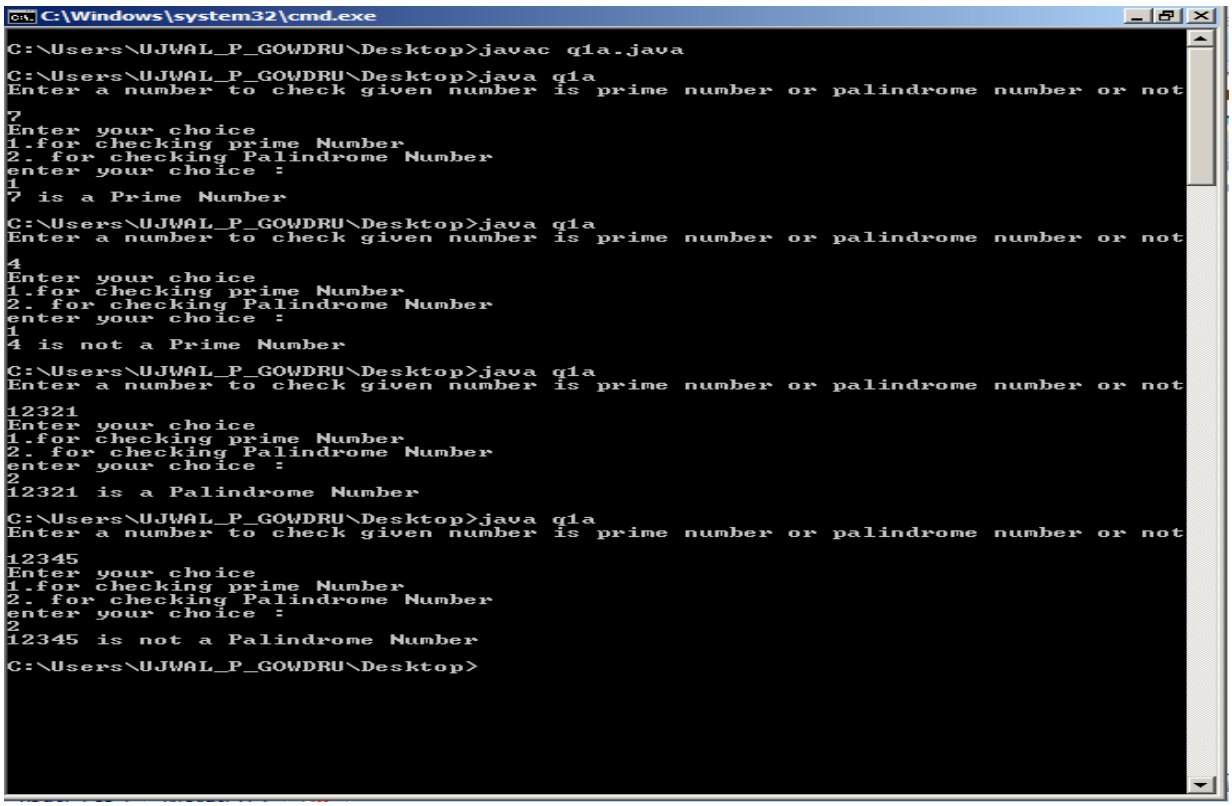
3) OOPs provide the ability to simulate real-world event much more effectively. We can provide the solution of real word problem if we are using the Object-Oriented Programming language.

1 a. Write a java program to check whether a given number is "prime number" and "palindrome number or not"

```
import java.util.Scanner;
import java.io.*;
public class q1a
 {
   public static void main(String[] args)
   {
      Scanner input = new Scanner(System.in);
      System.out.println("Enter a number to check given number is prime number or
      palindrome number or not");
      int num =input.nextInt();
      System.out.println("Enter your choice");
      System.out.println("1.for checking prime Number");
      System.out.println("2. for checking Palindrome Number");
      System.out.println("enter your choice :");
       int a= input.nextInt();
      switch (a)
              {
                case 1:
                        int flag=0;
                        for(int i=2;i<num;i++)
                           {
                            if(num%i==0)
                              {
                                System.out.println(num+" is not a Prime Number");
                                flag = 1;
                                 break;
                               }
                           }
                     if(flag==0)
                    System.out.println(num+" is a Prime Number");
                    break;
```

```
            case 2:
                    int k = num;
                    int reverse=0,rem;
                   while(num > 0)
                         {
                           rem = num % 10;
                           reverse = reverse * 10 + rem;
                           num = num / 10;
                         }
              if(reverse == k)
                 System.out.println(k+" is a Palindrome Number");
              else
                  System.out.println(k+" is not a Palindrome Number");
             break;
      }
   }
}
```

Output:



1 b. Write a java program to design a class to represent a bank account and include the following members. Date members are: Name of depositor, Account Number, Type of account and Balance amount in the account. Methods are: To assign initial values, to deposit an account, to withdraw an amount after checking balance and to display the name and balance.

```
import java.util.*;
class Pg1b
{
        String name,type;
        double accno,balance = 500.0;
        public void read()
        {
                Scanner sc = new Scanner(System.in);
                System.out.println("Enter Account Number : ");
                accno = sc.nextDouble();
                System.out.println("Enter Name of Depositor : ");
                name = sc.next();
                System.out.println("Enter Account Type : ");
                type = sc.next();
        }
        public void deposit(int amt)
        {
                double newbal = balance + amt;
                System.out.println("Balance Before Deposit : " + balance);
                System.out.println("Balance After Deposit : " + newbal);
                balance = newbal;
        }
        public void withdraw(int amt)
        {
                if(amt<=500 )
```

```
                {
                System.out.println("Withdraw is not possible because maintain minimum balance");
                        System.out.println("Available Balance is : " + balance);
                }
                else if (amt>balance)
                        {
                        System.out.println("Insufficient Balance");
                           System.out.println("Available Balance is : " + balance);
                }
                else
                {
                        double newbal = balance - amt;
                        System.out.println("Balance Before Withdrawal : " + balance);
                        System.out.println("Balance After Withdrawal : " + newbal);
                        balance = newbal;
          }
        }
        public void display()
 {
System.out.println("Account Holder Name : " + name);
System.out.println("Balance is : " + balance);
        }
        public static void main(String argsp[])
 {
        Pg1b b = new Pg1b();
  Scanner sc = new Scanner(System.in);
  int a,amt;
  do
  {
```

```
System.out.println("Select:\n1 - Enter Account Details \n2 - Deposit \n3 - Withdraw \n4 - Balance \n5 - Exit");
a = sc.nextInt();
switch(a)
{
        case 1:
 {
         b.read();
  break;
        }
        case 2:
        {
                System.out.println("Enter Deposit Amount : ");
                amt = sc.nextInt();
                b.deposit(amt);
                break;
        }
        case 3:
        {
                System.out.println("Enter Withdrawal Amount : ");
                amt = sc.nextInt();
                b.withdraw(amt);
                break;
        }
        case 4:
        {
                b.display();
                break;
        }
 }
}while(a!=5);
```

```
}

}
```

Output:

```
C:\Users\UJWAL_P_GOWDRU\Desktop>javac Pg1b.java

C:\Users\UJWAL_P_GOWDRU\Desktop>java Pg1b
Select:
1 - Enter Account Details
2 - Deposit
3 - Withdraw
4 - Balance
5 - Exit
1
Enter Account Number :
123
Enter Name of Depositor :
bsp
Enter Account Type :
saving
Select:
1 - Enter Account Details
2 - Deposit
3 - Withdraw
4 - Balance
5 - Exit
2
Enter Deposit Amount :
700
Balance Before Deposit : 500.0
Balance After Deposit : 1200.0
Select:
1 - Enter Account Details
2 - Deposit
3 - Withdraw
4 - Balance
5 - Exit
4
Account Holder Name : bsp
Balance is : 1200.0
Select:
1 - Enter Account Details
2 - Deposit
3 - Withdraw
4 - Balance
5 - Exit
3
Enter Withdrawal Amount :
600
```

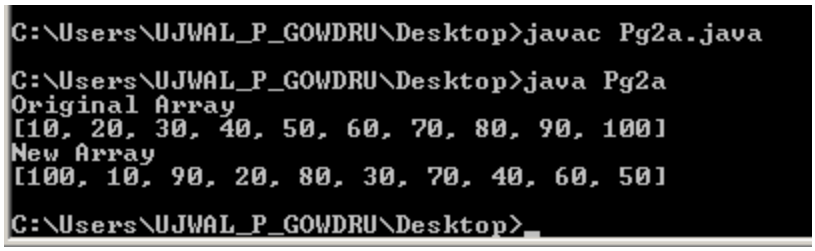
```
Balance Before Withdrawal : 1200.0
Balance After Withdrawal : 600.0
Select:
1 - Enter Account Details
2 - Deposit
3 - Withdraw
4 - Balance
5 - Exit
4
Account Holder Name : bsp
Balance is : 600.0
Select:
1 - Enter Account Details
2 - Deposit
3 - Withdraw
4 - Balance
5 - Exit
```

2 a. Write a Java program to sort an array of positive integers of a given array, in the sorted array the value of the first element should be maximum, second value should be minimum value, third should be second maximum, fourth second be second minimum and so on.

```
import java.util.Arrays;
public class Pg2a
{
    static int[] rearrange(int[] new_arra, int n)
   {
       int temp[] = new int[n];
        int small_num = 0, large_num = n-1;
        boolean flag = true;
        for (int i=0; i < n; i++)
       {
           if (flag)
               temp[i] = new_arra[large_num--];
           else
               temp[i] = new_arra[small_num++];
                flag = !flag;
       }
       return temp;
   }
    public static void main(String[] args)
   {
       int nums[] = new int[]{10, 20, 30, 40, 50, 60, 70, 80, 90, 100};
       int result[];
       System.out.println("Original Array ");
       System.out.println(Arrays.toString(nums));
       result = rearrange(nums,nums.length);
       System.out.println("New Array ");
       System.out.println(Arrays.toString(result));
```

```
    }
}
```

Output:


C:\Users\UJWAL_P_GOWDRU\Desktop>javac Pg2a.java
C:\Users\UJWAL_P_GOWDRU\Desktop>java Pg2a
Original Array
[10, 20, 30, 40, 50, 60, 70, 80, 90, 100]
New Array
[100, 10, 90, 20, 80, 30, 70, 40, 60, 50]
C:\Users\UJWAL_P_GOWDRU\Desktop>

2 b. 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**

```
import java.io.*;
import java.util.*;
class ElectricityBill
{
 double bill;
 double domesticbillcalc (int units)
 {
 if(units<100)
 bill = units * 1 ;
 else if(units <= 200)
 bill = 100 * 1 + (units - 100) * 2.50 ;
 else if(units <= 500)
 bill = 100 * 1 + 200 * 2.50 + (units - 200) * 4 ;
 else
```

```
 bill = 100 * 1 + 200 * 2.50 + 500 * 4 + (units - 500) * 6 ;
 return bill;
 }
 double commercialbillcalc (int units)
 {
 if(units<100)
 bill = units * 2 ;
 else if(units <= 200)
 bill = 100 * 1 + (units - 100) * 4.50 ;
 else if(units <= 500)
 bill = 100 * 1 + 200 * 4.50 + (units - 200) * 6 ;
 else
 bill = 100 * 1 + 200 * 4.50 + 500 * 6 + (units - 500) * 7 ;
 return bill;
 } void show(String ptype,String consno,String consname,int pmr,int cmr,int
units)
 {
 System.out.println("Type of Connection : " + ptype);
 System.out.println("Consumer Number : " + consno);
 System.out.println("Customer Name : " + consname);
 System.out.println("Previous Month Reading : " + pmr);
 System.out.println("Current Month Reading : " + cmr);
 System.out.println("Units Consumed : " + units);
 }
}
class Pg2b
{
 public static void main(String[] args)
 {
 Scanner c = new Scanner(System.in);
 System.out.println("Enter the Type of Connection :");
```

```
 String ptype=c.next();
 System.out.println("Enter the Consumer Number :");
 String consno=c.next();
 System.out.println("Enter the Consumer Name :");
 String consname=c.next();
 System.out.println("Enter the Previous Month Reading :");
 int pmr=c.nextInt();
System.out.println("Enter the Current Month Reading :");
 int cmr=c.nextInt();
 int units = cmr-pmr;
 ElectricityBill b = new ElectricityBill();
 if(ptype.equalsIgnoreCase("DOMESTIC"))
 {
 b.show(ptype,consno,consname,pmr,cmr,units);
 b.domesticbillcalc(units);
 System.out.println("Bill to pay : " + b.bill);
 }
 else if(ptype.equalsIgnoreCase("COMMERCIAL"))
 {
 b.show(ptype,consno,consname,pmr,cmr,units);
 b.commercialbillcalc(units);
 System.out.println("Bill to pay : " + b.bill);
 }
 }
}
```

Output:

```
C:\Users\UJWAL_P_GOWDRU\Desktop>javac Pg2b.java

C:\Users\UJWAL_P_GOWDRU\Desktop>java Pg2b
Enter the Type of Connection :
domestic
Enter the Consumer Number :
1
Enter the Consumer Name :
PUTTASWAMY
Enter the Previous Month Reading :
0
Enter the Current Month Reading :
550
Type of Connection : domestic
Consumer Number : 1
Customer Name : PUTTASWAMY
Previous Month Reading : 0
Current Month Reading : 550
Units Consumed : 550
Bill to pay : 2900.0
```

```
C:\Users\UJWAL_P_GOWDRU\Desktop>java Pg2b
Enter the Type of Connection :
domestic
Enter the Consumer Number :
1
Enter the Consumer Name :
asdf
Enter the Previous Month Reading :
100
Enter the Current Month Reading :
250
Type of Connection : domestic
Consumer Number : 1
Customer Name : asdf
Previous Month Reading : 100
Current Month Reading : 250
Units Consumed : 150
Bill to pay : 225.0
```

3 a. Write a program to read an array of n elements and implement following operations using Arrays class methods.

i) Perform Binary Search by reading element to be searched

ii) Initialize two arrays and check if the arrays are equal or not.

```
import java.util.Arrays;
import java.util.*;
public class Pg3a
{
   public static void main(String[] args)
   {
      int n,key,res;
      Boolean r;
      System.out.println("Enter number of Array Elements");
      Scanner inp = new Scanner(System.in);
      n=inp.nextInt();
      int arr[]=new int[n];
      System.out.println("Enter Array Elements");
      for(int i=0;i<n;i++)
         arr[i]=inp.nextInt();
      System.out.println("Array Elements entered are");
      for (int i=0;i<n;i++)
      System.out.println(arr[i]);
      System.out.println("Perform Binary Search");
      Arrays.sort(arr);
      System.out.println("Sorted Array");
      for (int i=0;i<n;i++)
      System.out.println(arr[i]);
      System.out.println("Enter Element to be searched");
      key=inp.nextInt();
      res=Arrays.binarySearch(arr,key);
      if(res>=0 && res<=n-1)
```

```
System.out.println(key + " found at index = " +res);
else
System.out.println(key + " not found");
int a[]={33,3,4,5};
int b[]={33,13,4,5};
System.out.println("Array A");
for (int x: a)
    System.out.println(x);
System.out.println("Array B");
for (int x: b)
    System.out.println(x);
r=Arrays.equals(a,b);
if(r==true)
  System.out.println("Arrays a and b are equal");
else
  System.out.println("Arrays a and b are not equal");
    }
}
```

Output:

3 b. A sales person is received commission based on the sales he makes as shown by the following table:

Sale	Commission
Up to ₹50,000	5% on sales value
More than ₹50,000 and up to ₹80,000	8% on sales value
More than ₹80,000 and up to ₹1,00,000	10% on sales value
More than ₹1,00,000	12% on sales value

Write a java program to define a class commission contains employee name, employee number and sales data member and methods input(), compute() and display() methods. Write a main method to create object of a class and call the above member methods.

```
import java.util.Scanner;
public class Commission
{
    String name;
    int emp_no;
    int sal;
    double comm;

    void input() {
        Scanner in = new Scanner(System.in);
        System.out.print("Enter employee name: ");
        name = in.nextLine();
        System.out.print("Enter employee number: ");
        emp_no = in.nextInt();
        System.out.print("Enter monthly sales value: ");
        sal = in.nextInt();
    }
      void compute()
      {
        if (sal <= 50000)
```

```
    {
        comm = 5.0 / 100.0 * sal;
    }
    else if (sal <= 80000)
    {
        comm = 8.0 / 100.0 * sal;
    }
    else if (sal <= 100000)
    {
        comm = 10.0 / 100.0 * sal;
    }
    Else
    {
        comm = 12.0 / 100.0 * sal;
    }
}

void display()
{
    System.out.println("Employee name: " + name);
    System.out.println("Employee Number: " + emp_no);
    System.out.println("Monthly Sales: " + sal);
    System.out.println("Commission: " + comm);
}

public static void main(String args[])
{
    Commission obj = new Commission();
    obj.input();
    obj.compute();
    obj.display();
```

```
    }
}
```

```
C:\Users\bsput\Desktop>javac Commission.java

C:\Users\bsput\Desktop>java Commission
Enter employee name: a
Enter employee number: 1
Enter monthly sales value: 0
Employee name: a
Employee Number: 1
Monthly Sales: 0
Commission: 0.0

C:\Users\bsput\Desktop>
```

4 a. Write a Java program to separate even and odd numbers of a given array of integers. Put all even numbers first, and then odd numbers. Also find the number of even and odd integers in a given array of integers. Array elements are user Inputs.

```
import java.util.Scanner;
import java.util.Arrays;
class Pg4a
{
 public static void main (String args[])
  {
    int counter=0;
    Scanner scan=new Scanner(System.in);  //create a scanner object for input
    System.out.print("Enter the array size :\n");
    int size=scan.nextInt();//reads input from user for array size
    System.out.print("Enter the elements of the array :\n");
    int arr[]=new int[size];
    for(int i=0; i<arr.length; i++)
     {
       arr[i]=scan.nextInt();
     }//reads input from user for array elements
System.out.print("Even numbers are: \n");
for(int i=0; i<size; i++)
    {
     if(arr[i]%2==0)
         {//separates even numbers
          System.out.println(arr[i]);
          counter++;
         }
        }
System.out.println("Number of even numbers : "+counter);
System.out.print("Odd numbers are: \n");
```

```
for(int i=0; i<size; i++)
    {
     if(arr[i]%2==1)
       {//separates odd numbers
         System.out.println(arr[i]);
        }
      }
System.out.println("Number of odd numbers  : "+(size-counter));
}
}
```

Output:

```
C:\Users\bsput\Desktop>javac Pg4a.java

C:\Users\bsput\Desktop>java Pg4a
Enter the array size :
5
Enter the elements of the array :
3
2
1
6
8
Even numbers are:
2
6
8
Number of even numbers  : 3
Odd numbers are:
3
1
Number of odd numbers   : 2

C:\Users\bsput\Desktop>
```

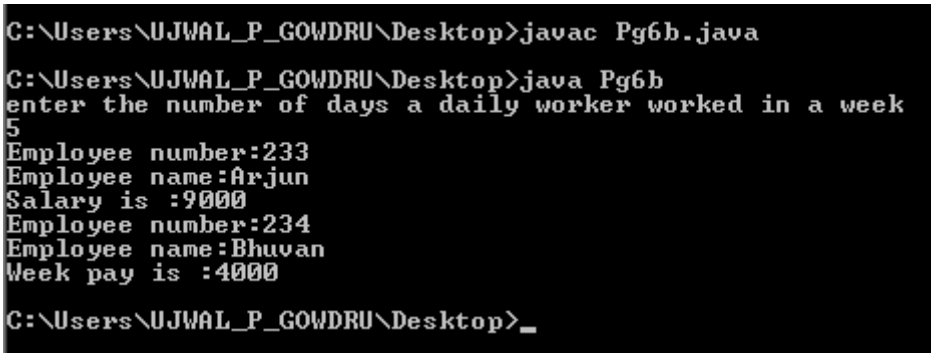
4 b. Write a class Worker and derive class DailyWorker and SalariedWorker from it. Every worker has a name and salary_rate. Write a method comPay(int hours) to compute the week pay of every worker. A daily worker is paid on the basis of number of days he work. The SalariedWorker gets paid the wage for 40hours a week no matter what actual hours is. Test this program to calculate the pay of workers. Write the program using the concept of polymorphism.

```
import java.io.*;
import java.util.Scanner;
class Worker
{
String name;
int empno;
Worker(int no, String n)
{
empno = no;
name = n;
}
void show()
{
System.out.println("Employee number:" + empno);
System.out.println("Employee name:" + name);
}
}// end of class 'Worker'
class DailyWorker extends Worker
{
int rate,hours;
DailyWorker(int no, String n, int r,int days)
{
super(no,n);
rate = r;
```

```
hours = days*24;
}
void compay()
{
show();
System.out.println("Salary is :" + rate*hours);
}
}// end of class 'DailyWorker'
class SalariedWorker extends Worker
{
int rate;
SalariedWorker(int no, String n, int r)
{
super(no,n);
rate = r;
}
void compay(int hours)
{
show();
System.out.println("Week pay is :" + rate*hours);
}
}// end of class 'SalariedWorker'
class Pg6b
{
public static void main(String[] args)
{
int days;
Scanner in = new Scanner(System.in);
System.out.println("enter the number of days a daily worker worked in a week");
days = in.nextInt();
DailyWorker d = new DailyWorker(233,"Arjun",75,days);
```

```
SalariedWorker s = new SalariedWorker(234,"Bhuvan",100);
d.compay();
s.compay(40);
} }
```

Output:



5 a. Write a java program to determine the addition of two matrices (using for loop).

```
import java.util.Scanner;
import java.io.*;
class AddMatrix
{
public static void main(String args[])
{
int row, col,i,j;
Scanner in = new Scanner(System.in);
System.out.println("Enter the number of rows");
row = in.nextInt();
System.out.println("Enter the number columns");
col = in.nextInt();
int mat1[][] = new int[row][col];
int mat2[][] = new int[row][col];
int res[][] = new int[row][col];
System.out.println("Enter the elements of matrix1");
 for ( i= 0 ; i < row ; i++ )
{
 for ( j= 0 ; j < col ;j++ )
mat1[i][j] = in.nextInt();
System.out.println();
}
System.out.println("Enter the elements of matrix2");
for ( i= 0 ; i < row ; i++ )
{
for ( j= 0 ; j < col ;j++ )
mat2[i][j] = in.nextInt();
System.out.println();
}
```

```
 for ( i= 0 ; i < row ; i++ )
for ( j= 0 ; j < col ;j++ )
res[i][j] = mat1[i][j] + mat2[i][j] ;
 System.out.println("Sum of matrices:-");
for ( i= 0 ; i < row ; i++ )
{
for ( j= 0 ; j < col ;j++ )
System.out.print(res[i][j]+"\t");
 System.out.println();
}
 }
}
```

Output:

```
Output                                                    Java
2
Enter the number  columns
2
Enter the elements of matrix1
1 1

1 1

Enter the elements of  matrix2
2 2

2 2

Sum of  matrices:-
3       3
3       3
```

5 b. Write a program to create an abstract class shape with two instance variables d(length of the side of polygon)and sides(no. of sides of a polygon),constructor and abstract method area. Create subclass polygon and circle. For polygon class create object references for triangle, square and hexagon. Invoke method area for objects of polygon and circle classes. Write constructors for subclasses.

```
abstract class shape
{
double d;
int sides;
shape(double d1,int s)
{
d=d1;
sides=s;
}
abstract double area();
}
class polygon extends shape
{
polygon(double d1,int s)
{
super(d1,s);
}
double area()
{
return sides*d*d/(4*Math.tan(3.14159/sides));
}
}
class circle extends shape
{
circle(double d1,int s)
```

```
{
super(d1,s);
}
double area()
{
return (3.14159*d*d*sides);
}}
class Pg7b
{
public static void main(String[] args)
{
circle c=new circle(10.0,1);
polygon t=new polygon(10.0,3);
polygon s=new polygon(10.0,4);
polygon h=new polygon(10.0,6);
System.out.println("Area of Circle="+c.area());
System.out.println("Area of Triangle="+t.area());
System.out.println("Area of Square="+s.area());
System.out.println("Area of Hexagon="+h.area());
}}
```

Output:

6 a. Write a java program to sort an array of strings in alphabetical order using array.sort() method.

```
import java.util.Arrays;
import java.util.Scanner;
public class Pg8a
{
  public static void main(String args[])
  {
    Scanner sc=new Scanner(System.in);
    int n;   //Declare the array size
    System.out.println("Enter the number of elements ");
    n=sc.nextInt();    //Initialize the array size
    String fruits[]=new String[n];   //Declare the array
    System.out.println("Enter the String ");
    Scanner sc1=new Scanner(System.in);
    for(int i=0; i<n ;i++)     //Initialize the array
    {
       fruits[i]=sc1.nextLine();
    }
      //logic for sorting
       for(int i = 0; i<n; i++)   //Holds each element
       {
          for (int j = i+1; j<n; j++)  //Check for remaining elements
          {
            //compares each elements of the array to all the remaining elements
            if(fruits[i].compareTo(fruits[j])>0)
            {
               //swapping array elements
               String temp = fruits[i];
               fruits[i] = fruits[j];
               fruits[j] = temp;
```

```
            }
          }
        }
        //prints the sorted array in alphabetical order
        System.out.println(Arrays.toString(fruits));
    }
}
```

Output:

```
C:\Users\UJWAL_P_GOWDRU\Desktop>javac Pg8a.java

C:\Users\UJWAL_P_GOWDRU\Desktop>java Pg8a
Enter the number of elements
5
Enter the String
UJWAL
PUTTA
NANDINI
SHIVALINGAMMA
MAHESH
[MAHESH, NANDINI, PUTTA, SHIVALINGAMMA, UJWAL]

C:\Users\UJWAL_P_GOWDRU\Desktop>
```

6 b. 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.

```
import java.io.*;
import java.util.*;
class Pg11a
{
public static void main(String[] args){
Scanner input = new Scanner(System.in);
System.out.println("Enter the name of the file:");
String file_name = input.nextLine();
File f = new File(file_name);
if(f.exists())
System.out.println("The file " +file_name+ " exists");
else
System.out.println("The file " +file_name+ " does not exist");
if(f.exists()){
if(f.canRead())
System.out.println("The file " +file_name+ " is readable");
else
System.out.println("The file " +file_name+ " is not readable");
if(f.canWrite())
System.out.println("The file " +file_name+ " is writeable");
else
System.out.println("The file " +file_name+ " is not writeable");
System.out.println("The file type is: " +file_name.substring(file_name.indexOf('.')+1));
System.out.println("The Length of the file:" +f.length());
}
```

```
}

}
```

Output:

```
F:\bsp java programs>javac Pg11a.java

F:\bsp java programs>java Pg11a
Enter the name of the file:
123.doc
The file 123.doc does not exist

F:\bsp java programs>java Pg11a
Enter the name of the file:
x.txt
The file x.txt does not exist

F:\bsp java programs>java Pg11a
Enter the name of the file:
a.txt
The file a.txt exists
The file a.txt is readable
The file a.txt is writeable
The file type is: txt
The Length of the file:42

F:\bsp java programs>
```

7 a. Java Program to Find Largest Between Three Numbers Using Ternary Operator

```
import java.util.Scanner;
public class Pg9a
{
    public static void main(String[] args)
    {
        int a, b, c, d;
        Scanner s = new Scanner(System.in);
        System.out.println("Enter all three numbers:");
        a = s.nextInt();
        b = s.nextInt();
        c = s.nextInt();
        d = c > (a > b ? a : b) ? c : ((a > b) ? a : b);
        System.out.println("Largest Number:"+d);
    }
}
```

Output:

```
C:\Users\UJWAL_P_GOWDRU\Desktop>javac Pg9a.java

C:\Users\UJWAL_P_GOWDRU\Desktop>java Pg9a
Enter all three numbers:
23
33
12
Largest Number:33

C:\Users\UJWAL_P_GOWDRU\Desktop>
```

7 b. Write a java application to generate pay slip for different category of employees using the concept of inheritance

```
import java.util.*;
class employee
{
 int empid;
 String name, address;
Scanner get = new Scanner(System.in);
void getdata()
{
 System.out.println("Enter Name of the Employee");
 name = get.nextLine();
  System.out.println("Enter Address of the Employee:");
 address = get.nextLine();
 System.out.println("Enter employee id ");
 empid = get.nextInt();
  }
 void display()
 {
 System.out.println("Employee Name: "+name);
 System.out.println("Employee id : "+empid);
  System.out.println("Address: "+address);
 }
}
class asstprofessor extends employee
{
double salary,bp,da,hra,pf,club,net,gross;
void getasst()
{
System.out.println("Enter basic pay");
bp = get.nextDouble();
```

```
}
void calculateasst()
{
da=(0.97*bp);
hra=(0.10*bp);
pf=(0.12*bp);
club=(0.1*bp);
gross=(bp+da+hra);
net=(gross-pf-club);
System.out.println("***********************************************");
System.out.println("PAY SLIP FOR ASSISTANT PROFESSOR");
System.out.println("***********************************************");
System.out.println("Basic Pay:Rs"+bp);
System.out.println("DA:Rs"+da);
System.out.println("HRA:Rs"+hra);
System.out.println("PF:Rs"+pf);
System.out.println("CLUB:Rs"+club);
System.out.println("GROSS PAY:Rs"+gross);
System.out.println("NET PAY:Rs"+net);
 }
}
class associateprofessor extends employee
{
double salary,bp,da,hra,pf,club,net,gross;
 void getassociate()
{
System.out.println("Enter basic pay");
bp = get.nextDouble();
}
void calculateassociate()
{
```

```
da=(0.97*bp);
hra=(0.10*bp);
pf=(0.12*bp);
club=(0.1*bp);
gross=(bp+da+hra);
net=(gross-pf-club);
System.out.println("*************************************************");
System.out.println("PAY SLIP FOR ASSOCIATE PROFESSOR");
System.out.println("*************************************************");
System.out.println("Basic Pay:Rs"+bp);
System.out.println("DA:Rs"+da);
System.out.println("HRA:Rs"+hra);
System.out.println("PF:Rs"+pf);
System.out.println("CLUB:Rs"+club);
System.out.println("GROSS PAY:Rs"+gross);
System.out.println("NET PAY:Rs"+net);
}
}
class professor extends employee
{
double salary,bp,da,hra,pf,club,net,gross;
void getprofessor()
{
System.out.println("Enter basic pay");
bp = get.nextDouble();
}
void calculateprofessor()
{
da=(0.97*bp);
hra=(0.10*bp);
pf=(0.12*bp);
```

```
club=(0.1*bp);
gross=(bp+da+hra);
net=(gross-pf-club);
System.out.println("************************************************");
System.out.println("PAY SLIP FOR PROFESSOR");
System.out.println("************************************************");
System.out.println("Basic Pay:Rs"+bp);
System.out.println("DA:Rs"+da);
System.out.println("HRA:Rs"+hra);
System.out.println("PF:Rs"+pf);
System.out.println("CLUB:Rs"+club);
System.out.println("GROSS PAY:Rs"+gross);
System.out.println("NET PAY:Rs"+net);
}
}
class Pg9b
{
public static void main(String args[])
{
int choice,cont;
do
{
System.out.println("PAYROLL");
System.out.println(" 1.ASSISTANT PROFESSOR \t 2.ASSOCIATE PROFESSOR
      \t 3.PROFESSOR ");
Scanner c = new Scanner(System.in);
choice=c.nextInt();
switch(choice)
{
case 1:
{
```

```
asstprofessor asst=new asstprofessor();
asst.getdata();
asst.getasst();
asst.display();
asst.calculateasst();
break;
}
case 2:
{
associateprofessor asso=new associateprofessor();
asso.getdata();
asso.getassociate();
asso.display();
asso.calculateassociate();
break;
}
case 3:
{
professor prof=new professor();
prof.getdata();
prof.getprofessor();
prof.display();
prof.calculateprofessor();
break;
}
}
System.out.println("Do u want to continue 0 to quit and  1 to continue ");
cont=c.nextInt();
}while(cont==1);
}
}
```

Output:

```
C:\Users\UJWAL_P_GOWDRU\Desktop>javac Pg9b.java

C:\Users\UJWAL_P_GOWDRU\Desktop>java Pg9b
PAYROLL
 1.ASSISTANT PROFESSOR   2.ASSOCIATE PROFESSOR   3.PROFESSOR
1
Enter Name of the Employee
puttaswamy
Enter Address of the Employee:
mandya
Enter employee id
1234
Enter basic pay
15600
Employee Name: puttaswamy
Employee id : 1234
Address: mandya
**************************************************
PAY SLIP FOR ASSISTANT PROFESSOR
**************************************************
Basic Pay:Rs15600.0
DA:Rs15132.0
HRA:Rs1560.0
PF:Rs1872.0
CLUB:Rs1560.0
GROSS PAY:Rs32292.0
NET PAY:Rs28860.0
Do u want to continue 0 to quit and  1 to continue
```

8 a. Write a Java program to display the Fibonacci series

```
import java.util.Scanner;
public class Pg10a
{
  public static void main(String[] args)
{
        int count, num1 = 0, num2 = 1;
        System.out.println("How may numbers you want in the sequence:");
        Scanner scanner = new Scanner(System.in);
        count = scanner.nextInt();
        scanner.close();
        System.out.print("Fibonacci Series of "+count+" numbers:");

        int i=1;
        while(i<=count)
        {
            System.out.print(num1+" ");
            int sumOfPrevTwo = num1 + num2;
            num1 = num2;
            num2 = sumOfPrevTwo;
            i++;
        }
    }
}
```

Output:

```
C:\Users\UJWAL_P_GOWDRU\Desktop>javac Pg10a.java

C:\Users\UJWAL_P_GOWDRU\Desktop>java Pg10a
How may numbers you want in the sequence:
5
Fibonacci Series of 5 numbers:0 1 1 2 3
C:\Users\UJWAL_P_GOWDRU\Desktop>^A
```

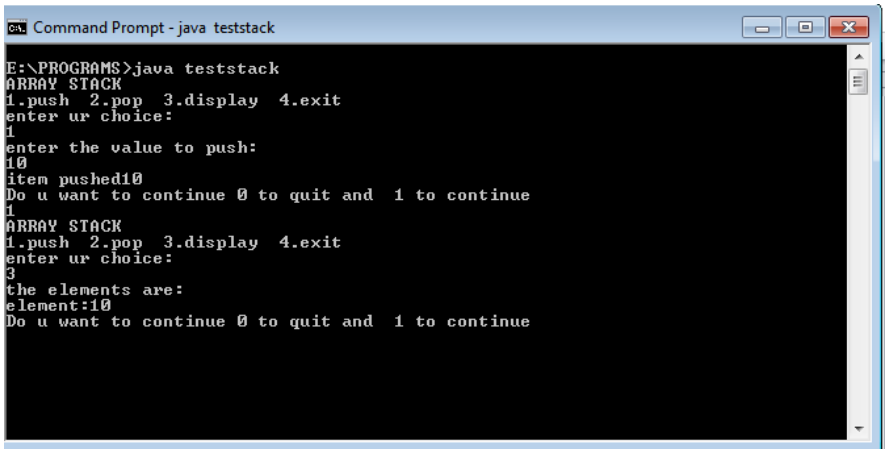
8 b. Write a java program for array implementation of stack using the concept of interface and exception handling.

```
import java.io.*;
interface stackoperation
{
 public void push(int i);
 public void pop();
}
class Astack implements stackoperation
{
int stack[]=new int[5];
int top=-1;
int i;
public void push(int item)
{
if(top>=4)
{
System.out.println("overflow");
}
else
{
top=top+1;
stack[top]=item;
System.out.println("item pushed"+stack[top]);
}
}
public void pop()
{
if(top<0)
System.out.println("underflow");
else
```

```
{
System.out.println("item popped"+stack[top]);
top=top-1;
}
}
public void display()
{
if(top<0)
System.out.println("No Element in stack");
else
{
for(i=0;i<=top;i++)
System.out.println("element:"+stack[i]);
}
}
}
class Pg10b
{
public static void main(String args[])throws IOException
{
int ch,c;
int i;
Astack s=new Astack();
DataInputStream in=new DataInputStream(System.in);
do
{
try
{
System.out.println("ARRAY STACK");
System.out.println("1.push  2.pop  3.display  4.exit");
System.out.println("enter ur choice:");
```

```
ch=Integer.parseInt(in.readLine());
switch(ch)
{
case 1:
System.out.println("enter the value to push:");
i=Integer.parseInt(in.readLine());
s.push(i);
break;
case 2:
s.pop();
break;
case 3:
System.out.println("the elements are:");
s.display();
break;
case 4:
break;
}
}
catch(IOException e)
{
System.out.println("io error");
}
System.out.println("Do u want to continue 0 to quit and  1 to continue ");
c=Integer.parseInt(in.readLine());
}while(c==1);
}
}
```

Output:


Command Prompt - java teststack
E:\PROGRAMS>java teststack
ARRAY STACK
1.push 2.pop 3.display 4.exit
enter ur choice:
1
enter the value to push:
10
item pushed10
Do u want to continue 0 to quit and 1 to continue
1
ARRAY STACK
1.push 2.pop 3.display 4.exit
enter ur choice:
3
the elements are:
element:10
Do u want to continue 0 to quit and 1 to continue

9 a. Write a program to read a string. Replace a character at the specified index in the String in Java

```
import java.util.Scanner;
public class Pg11a
{
public static void main(String[] args)
{
        String str ;
        Scanner in = new Scanner(System.in);
System.out.println("Enter a string");
str  =in.nextLine();
System.out.println("Enter a valid index");
int index;
index=in.nextInt();
System.out.println("Enter a character to replace at the index value");
char ch;
ch=in.next().charAt(0) ;
System.out.println("Original String = "+ str);
str = str.substring(0, index) + ch+ str.substring(index + 1);
System.out.println("Modified String = "+ str);
    }
}
```

Output:

```
C:\Users\UJWAL_P_GOWDRU\Desktop>javac Pg11a.java

C:\Users\UJWAL_P_GOWDRU\Desktop>java Pg11a
Enter a string
Information Science and Engineering
Enter a valid index
0
Enter a character to replace at the index value
e
Original String = Information Science and Engineering
Modified String = enformation Science and Engineering

C:\Users\UJWAL_P_GOWDRU\Desktop>
```

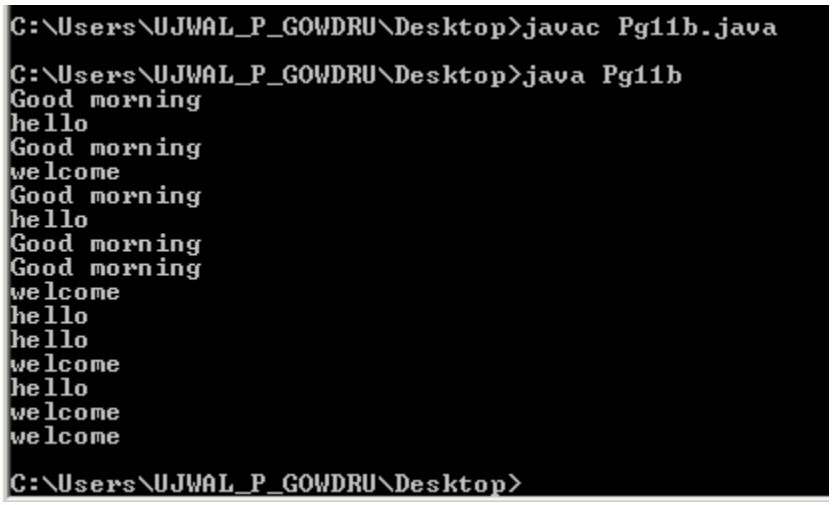
9 b. Write a JAVA program that creates threads by extending Thread class First thread display "Good Morning "every 1 sec, the second thread displays "Hello "every 2 seconds and the third display "Welcome" every 3 seconds

```
class GoodMorning extends Thread
{
   synchronized public void run()
{
       try
{
           int i=0;
           while (i<5)
{
               sleep(1000);
               System.out.println("Good morning ");
               i++;
           }
       }
catch (Exception e)
{
       }
   }
}
 class Hello extends Thread
{
   synchronized public void run()
{
       try
{
           int i=0;
           while (i<5) {
```

```
                sleep(2000);
                System.out.println("hello");
                i++;
            }
        }
catch (Exception e)
{
        }
    }
}
class Welcome extends Thread
{
    synchronized public void run()
{
        try {
            int i=0;
            while (i<5) {
                sleep(3000);
                System.out.println("welcome");
                i++;
            }
        }
catch (Exception e)
{
        }
    }
}
class Pg11b
{
    public static void main(String args[])
{
```

```
        GoodMorning t1 = new GoodMorning();
        Hello t2 = new Hello();
        Welcome t3 = new Welcome();
        t1.start();
        t2.start();
        t3.start();
    }
}
```

Output:



10 a. Write a Java Program to find all substrings of a given string.

```
import java.util.Scanner;
public class Pg7a
{
public static void main(String[] args)
{
  String str, substr;
   int i, c, len;
   System.out.println("Enter a string to print all its substrings");
   Scanner in = new Scanner(System.in);
   str = in.nextLine();
   len = str.length();
   System.out.println("Substrings of " + str + " are:");
   for (c = 0; c <len; c++) //c points to index of first character in string
        {
          for (i = 1; i<= len - c; i++)
                {
                 substr = str.substring(c, c + i);
                 System.out.println(substr);
                }
        }
    }
}
```

Output:

```
C:\Users\UJWAL_P_GOWDRU\Desktop>javac Pg7a.java

C:\Users\UJWAL_P_GOWDRU\Desktop>java Pg7a
Enter a string to print all its substrings
PESCE
Substrings of PESCE are:
P
PE
PES
PESC
PESCE
E
ES
ESC
ESCE
S
SC
SCE
C
CE
E

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

10 b. Write a Java program to implement currency converter, distance converter and time converter using packages

```
package currencyconversion;
import java.util.*;
public class currency
{
double inr,usd;
double euro,yen;
Scanner in=new Scanner(System.in);
public void dollartorupee()
{
System.out.println("Enter dollars to convert into Rupees:");
usd=in.nextInt();
inr=usd*67;
System.out.println("Dollar ="+usd+"equal to INR="+inr);
}
public void rupeetodollar()
{
System.out.println("Enter Rupee to convert into Dollars:");
inr=in.nextInt();
usd=inr/67;
System.out.println("Rupee ="+inr+"equal to Dollars="+usd);
}
public void eurotorupee()
{
System.out.println("Enter euro to convert into Rupees:");
euro=in.nextInt();
inr=euro*79.50;
```

```
System.out.println("Euro ="+euro +"equal to INR="+inr);
}
public void rupeetoeuro()
{
System.out.println("Enter Rupees to convert into Euro:");
inr=in.nextInt();
euro=(inr/79.50);
System.out.println("Rupee ="+inr +"equal to Euro="+euro);
}
public void yentorupee()
{
System.out.println("Enter yen to convert into Rupees:");
yen=in.nextInt();
inr=yen*0.61;
System.out.println("YEN="+yen +"equal to INR="+inr);
}
public void rupeetoyen()
{
System.out.println("Enter Rupees to convert into Yen:");
inr=in.nextInt();
yen=(inr/0.61);
System.out.println("INR="+inr +"equal to YEN"+yen);
}
}
```

distance.java

```
package distanceconversion;
import java.util.*;
public class distance
```

```
{
double km,m,miles;
Scanner sc = new Scanner(System.in);
public void kmtom()
{
System.out.print("Enter in km ");
km=sc.nextDouble();
m=(km*1000);
System.out.println(km+"km" +"equal to"+m+"metres");
}
public void mtokm()
{
System.out.print("Enter in meter ");
m=sc.nextDouble();
km=(m/1000);
System.out.println(m+"m" +"equal to"+km+"kilometres");
}
public void milestokm()
{
System.out.print("Enter in miles");
miles=sc.nextDouble();
km=(miles*1.60934);
System.out.println(miles+"miles" +"equal to"+km+"kilometres");
}
public void kmtomiles()
{
System.out.print("Enter in km");
km=sc.nextDouble();
```

```
miles=(km*0.621371);
System.out.println(km+"km" +"equal to"+miles+"miles");
}
}
```

timer.java

```
package timeconversion;
import java.util.*;
public class timer
{
int hours,seconds,minutes;
int input;
Scanner sc = new Scanner(System.in);
public void secondstohours()
{
System.out.print("Enter the number of seconds: ");
input = sc.nextInt();
hours = input / 3600;
minutes = (input % 3600) / 60;
seconds = (input % 3600) % 60;
System.out.println("Hours: " + hours);
System.out.println("Minutes: " + minutes);
System.out.println("Seconds: " + seconds);
}
public void minutestohours()
{
System.out.print("Enter the number of minutes: ");
minutes=sc.nextInt();
hours=minutes/60;
```

```
minutes=minutes%60;
System.out.println("Hours: " + hours);
System.out.println("Minutes: " + minutes);
}
public void hourstominutes()
{
System.out.println("enter the no of hours");
hours=sc.nextInt();
minutes=(hours*60);
System.out.println("Minutes: " + minutes);
}
public void hourstoseconds()
{
System.out.println("enter the no of hours");
hours=sc.nextInt();
seconds=(hours*3600);
System.out.println("Minutes: " + seconds);
}
}
```

converter.java

```
import java.util.*;
import java.io.*;
import currencyconversion.*;
import distanceconversion.*;
import timeconversion.*;
class converter
{
public static void main(String args[])
```

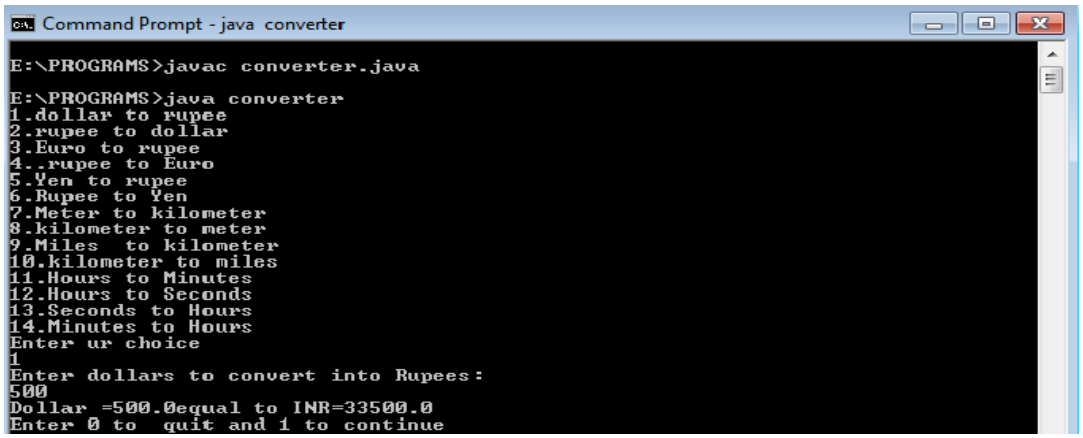
```
{
Scanner s=new Scanner(System.in);
int choice,ch;
currency c=new currency();
distance d=new distance();
timer t=new timer();
do
{
System.out.println("1.dollar to rupee ");
System.out.println("2.rupee to dollar ");
System.out.println("3.Euro to rupee ");
System.out.println("4..rupee to Euro ");
System.out.println("5.Yen to rupee ");
System.out.println("6.Rupee to Yen ");
System.out.println("7.Meter to kilometer ");
System.out.println("8.kilometer to meter ");
System.out.println("9.Miles  to kilometer ");
System.out.println("10.kilometer to miles");
System.out.println("11.Hours to Minutes");
System.out.println("12.Hours to Seconds");
System.out.println("13.Seconds to Hours");
System.out.println("14.Minutes to Hours");
System.out.println("Enter ur choice");
choice=s.nextInt();
switch(choice)
{
case 1:
{
```

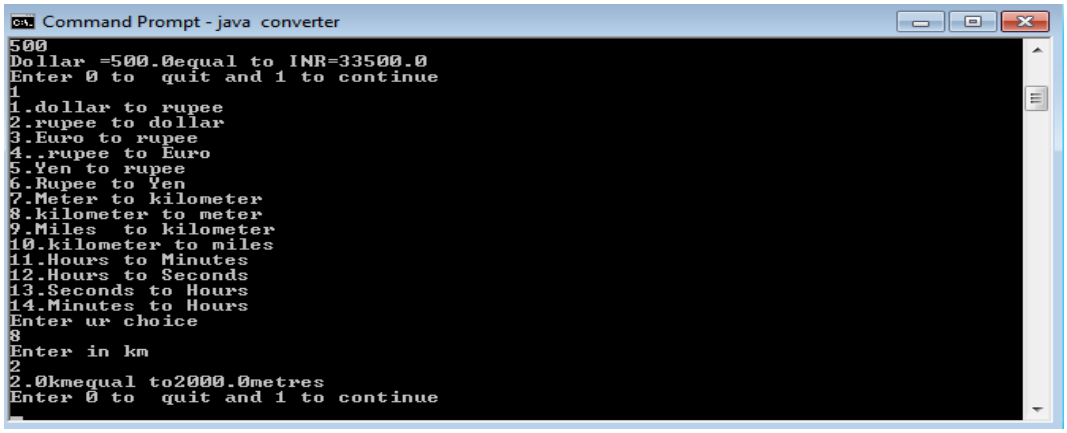
```
c.dollartorupee();
break;
}
case 2:
{
c.rupeetodollar();
break;
}
case 3:
{
c.eurotorupee();
break;
}
case 4:
{
c.rupeetoeuro();
break;
}
case 5:
{c.yentorupee();
break;}
case 6 :
{
c.rupeetoyen();
break;
}
case 7  :
{
```

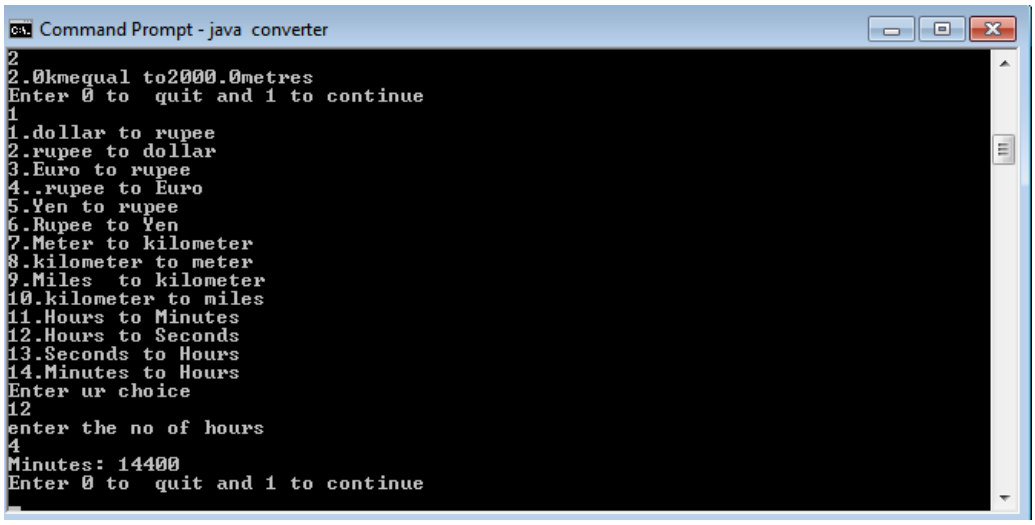
```
d.mtokm();
break;
}
case 8  :
{
d.kmtom();
break;
}
case 9  :
{
d.milestokm();
break;
}
case 10  :
{
d.kmtomiles();
break;
}
case 11 :
{
t.hourstominutes();
break;
}
case 12 :
{
t.hourstoseconds();
break;
}
```

```
case 13 :
{
t.secondstohours();
break;
}
case 14 :
{
t.minutestohours();
break;
}}
System.out.println("Enter 0 to  quit and 1 to continue ");
ch=s.nextInt();
}while(ch==1);
}
}
```

Output:




Command Prompt - java converter
500
Dollar =500.0equal to INR=33500.0
Enter 0 to quit and 1 to continue
1
1.dollar to rupee
2.rupee to dollar
3.Euro to rupee
4..rupee to Euro
5.Yen to rupee
6.Rupee to Yen
7.Meter to kilometer
8.kilometer to meter
9.Miles to kilometer
10.kilometer to miles
11.Hours to Minutes
12.Hours to Seconds
13.Seconds to Hours
14.Minutes to Hours
Enter ur choice
8
Enter in km
2
2.0kmequal to2000.0metres
Enter 0 to quit and 1 to continue


Command Prompt - java converter
2
2.0kmequal to2000.0metres
Enter 0 to quit and 1 to continue
1
1.dollar to rupee
2.rupee to dollar
3.Euro to rupee
4..rupee to Euro
5.Yen to rupee
6.Rupee to Yen
7.Meter to kilometer
8.kilometer to meter
9.Miles to kilometer
10.kilometer to miles
11.Hours to Minutes
12.Hours to Seconds
13.Seconds to Hours
14.Minutes to Hours
Enter ur choice
12
enter the no of hours
4
Minutes: 14400
Enter 0 to quit and 1 to continue