

*A Project report*

*On*

# **SAFE SHARING: ACCESS CONTROL FOR CLOUD STORED DATA**

*Submitted in partial fulfillment of the requirements  
for the award of the degree of*

**BACHELOR OF TECHNOLOGY**

*in*

**Computer Science & Engineering**

*By*

<b>G. AJAY KISHORE</b>	<b>204G1A0506</b>
<b>M. MOUNIKA</b>	<b>204G1A0561</b>
<b>T. HARSHA SRI</b>	<b>204G1A0536</b>
<b>B. BHAVANA</b>	<b>204G1A0522</b>

Under the Guidance of

**Mr. M. Narasimhulu** M.Tech,(Ph.D)

Assistant Professor



**Department of Computer Science & Engineering**

**SRINIVASA RAMANUJAN INSTITUTE OF TECHNOLOGY**  
(AUTONOMOUS)

(Affiliated to JNTUA, Accredited by NAAC with 'A' Grade, Approved by AICTE, New Delhi &  
Accredited by NBA (EEE, ECE & CSE))

Rotarypuram Village, BK Samudram Mandal, Ananthapuramu-515701

**2023-2024**

# SRINIVASA RAMANUJAN INSTITUTE OF TECHNOLOGY

(AUTONOMOUS)

(Affiliated to JNTUA, Accredited by NAAC with 'A' Grade, Approved by AICTE, New Delhi & Accredited by NBA (EEE, ECE & CSE))

Rotarypuram Village, BK Samudram Mandal, Ananthapuramu-515701

## DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING



### Certificate

This is to certify that the Project report entitled **SAFE SHARING: ACCESS CONTROL FOR CLOUD STORED DATA** is the bonafide work carried out by **G. AJAY KISHORE, M. MOUNIKA, T. HARSHA SRI, B. BHAVANA** bearing Roll Number **204G1A0506, 204G1A0561, 204G1A0536, 204G1A0522** in partial fulfilment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science & Engineering** during the academic year 2023 - 2024.

#### **Project Guide**

Mr. M. Narasimhulu M.Tech.,(Ph.D)  
Assistant Professor

#### **Head of the Department**

Mr. P.Veera Prakash M.Tech.,(Ph.D.)  
Assistant Professor & HOD

Date:

**EXTERNAL EXAMINER**

Place: Rotarypuram

## ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of people who made it possible, whose constant guidance and encouragement crowned our efforts with success. It is a pleasant aspect that we have now the opportunity to express our gratitude for all of them.

It is with immense pleasure that we would like to express our indebted gratitude to our Guide **Mr. M. Narasimhulu, M.Tech,(PhD)., Assistant Professor, Computer Science & Engineering**, who has guided us a lot and encouraged us in every step of the project work. We thank him for the stimulating guidance, constant encouragement and constructive criticism which have made possible to bring out this project work.

We are very much thankful to **Mr. P. Veera Prakash, M.Tech., (Ph.D.), Assistant Professor & Head of the Department, Computer Science & Engineering**, for his kind support and for providing necessary facilities to carry out the work.

We wish to convey our special thanks to **Dr. G. Bala Krishna, Ph.D., Principal of Srinivasa Ramanujan Institute of Technology** for giving the required information in doing our project work. Not to forget, We thank all other faculty and non- teaching staff, and our friends who had directly or indirectly helped and supported us in completing our project in time.

We also express our sincere thanks to the Management for providing excellent facilities.

Finally, we wish to convey our gratitude to our families who fostered all the requirements and facilities that we need.

### Project Associates

204G1A0506	AJAY KISHORE G
204G1A0561	MOUNIKA M
204G1A0536	HARSHA SRI T
204G1A0522	BHAVANA B

## DECLARATION

We, Mr G. Ajay kishore with reg no: 204G1A0506, Ms M. MOUNIKA with reg no: 204G1A0561, Ms T. Harsha sri with reg no: 204G1A0536 , Ms B. Bhavana with reg no: 204G1A0522 students of SRINIVASA RAMANUJAN INSTITUTE OF TECHNOLOGY , Rotarypuram , hereby declare that the dissertation entitled “**SAFE SHARING: ACCESS CONTROL FOR CLOUD STORED DATA** ” embodies the report of our project work carried out by us during IV year Bachelor of Technology under the guidance of **Mr. M. Narasimhulu** M.Tech.,(PhD). Department of CSE, **SRINIVASA RAMANUJAN INSTITUTE OF TECHNOLOGY**, and this work has beensubmitted for the partial fulfillment of the requirements for the award of the Bachelor of Technology degree.

The results embodied in this project have not been submitted to any other University of Institute for the award of any Degree or Diploma.

G. AJAY KISHORE

Reg no: 204G1A0506

M. MOUNIKA

Reg no: 204G1A0561

T. HARSHA SRI

Reg no: 204G1A0536

B. BHAVANA

Reg no: 204G1A0522

	<b>CONTENTS</b>	<b>Page No.</b>
<b>List of Figures</b>		<b>VII</b>
<b>Abbreviations</b>		<b>VIII</b>
<b>Abstract</b>		<b>IX</b>
<b>Chapter 1</b>	<b>Introduction</b>	<b>1</b>
	1.1 Problem Statement	2
	1.2 Objectives	3
	1.3 Scope of Project	3
<b>Chapter 2</b>	<b>Literature Survey</b>	<b>4</b>
<b>Chapter 3</b>	<b>System Analysis</b>	<b>7</b>
	3.1 Architecture	7
	3.2 Software Development Life Cycle	10
	3.3 Feasibility Study	
<b>Chapter 4</b>	<b>System Requirements Specification</b>	<b>12</b>
	4.1 Functional Requirements	13
	4.2 Non-Functional Requirements	14
	4.5 Hardware Requirements	19
	4.6 Software Requirements	22
<b>Chapter 5</b>	<b>Design</b>	<b>25</b>
	5.1 Input Design	25
	5.1.1 Objectives for Input Design	29
	5.2 Output Design	29
	5.3 Modules	30

	5.4 UML Diagrams	<b>32</b>
	5.5 Use Case Diagrams	34
	5.6 Class Diagram	35
	5.7 Sequence Digarams	
	5.8 Collaboration Diagram	
	5.9 Activity Diagram	35
	5.10 Component Diagram	36
	5.11 Deployment Diagram	36
<b>Chapter 6</b>	<b>Software Environment and Installation</b>	<b>37</b>
<b>Chapter 7</b>	<b>Output</b>	40
<b>Conclusion</b>		46
<b>References</b>		47

## **List of Figures**

<b>Fig No.</b>	<b>Description</b>	<b>PageNo</b>
Fig.3.1	Architecture	8
Fig.3.2	Waterfall Model	9
Fig.5.5	Use case Diagram	9
Fig.5.6	Class Diagram	10
Fig.5.7	Sequence Diagram	11
Fig.5.8	Collaboration Diagram	13
Fig.5.10	Activity Diagram	20
Fig.5.11	Deployment Diagram	20
Fig.6.1	Java Byte Code	21
Fig.6.2	Java Compilation	22
Fig.6.3	Java Platform	23
Fig.6.4	Java IDE	23
Fig.6.5	OSI Model	24
Fig.6.6	IP Address and Subnet	
Fig.6.7	Java JDK	27
Fig.6.8	Versions of Java	27
Fig.6.9	Java Downloads	28
Fig.6.10	JDK Setup	29
Fig.6.11	System Architecture	30
Fig.6.12	Path Selection	31
Fig.6.13	Java Dialog Box	32
Fig.6.14	Window Properties	33
Fig.6.15	Environment Variables	
Fig.6.16	Buttons of Variable Users	
Fig.6.17	Path in the Variable name	41
Fig.6.18	Copying the Path	41
Fig.6.19	Pasting the Path	42
Fig.6.20	Set Class Path	42
Fig.6.21	Setting Environment Variables	43
Fig.6.22	Java Command prompt	
Fig.6.23	Eclipse Packages	44
Fig.6.24	Eclipse Download	44
Fig.6.25	Eclipse Zip Download	45
Fig.6.26	File Appearance	45

## **LIST OF ABBREVIATIONS**

SSE	Searchable Symmetric Encryption
ABE	Attribute Based Encryption
CSP	Cloud Service Provider
RBAC	Role Based Access Control
SDLC	Software Development Life Cycle
SRS	System Requirements Specification
TCP	Transmission Control Protocol
UML	Unified Modelling Language



## ABSTRACT

The rapid expansion of cloud environments has brought about a significant challenge to secure data storage. This is a critical consideration for every user when decided to move the data online. To address this challenge, various solutions have been proposed, with two prominent approaches being Searchable Symmetric Encryption and Attribute-Based Encryption. SSE offers protection against both external and internal threats. It allows for efficient search capabilities while maintaining the confidentiality of the data. In an SSE technique, all data is usually encrypted using a single key. The entire encrypted database would need to be downloaded and re-encrypted with a new key if a user was to be revoked. Conversely, though, ABE offers a more granular approach to access control by encrypting data based on attributes and policies. This means that different users or groups can be granted different levels of access to the data based on their attributes.

**Keywords:** Encryption, Cryptography, Access Control, Searchable Symmetric Encryption (SSE), Attribute-Based Encryption(ABE)

# CHAPTER 1

## INTRODUCTION

People may rapidly and simply execute crucial tasks with their data in cloud computing, such as locating, transferring, and conserving it. However, maintaining the security of the data is a challenge. This is because the information is maintained by a different organization, and poorly protected data carries the highest risks.

The last several years have seen such rapid development in cloud computing that almost everyone's everyday life is now significantly impacted by it. The cloud is currently used on a daily basis by both large corporations and regular internet users. Many users are still hesitant to outsource their personal information, though, because cloud services are housed and managed by questionable third parties, making the contents susceptible to internal attacks.

Major players in the business as well as researchers have looked at attribute-based and symmetric searchable encryption as potential solutions for this reason. Before sending their files to the Cloud Service Provider (CSP), users in an SSE system encrypt them locally. As a result, the CSP that does not have the encryption key cannot obtain any relevant information about the users' data. The ability to do a direct keyword search on encrypted data is the most exciting feature of SSE, though. Unfortunately, user revocation is not supported by SSE systems, which is a major problem for cloud-based apps. Thus, eliminating a user corresponds to download the entire database and again encrypting it using a new key.

An alternative approach that functions in cloud-based applications is ABE. A master public key is used to encrypt every file in ABE schemes; however, Unlike conventional public key cryptosystems, which use the ciphertext that is generated is limited by a policy. Every user also has a secret key which is unique and associated with the users attributes. As a result a file can only be unlocked if and when the user's attributes align with the ciphertext's policy. However, encrypting vast amounts of data with an asymmetric encryption method is not particularly effective.

Additionally, the solution must ensure secure authentication and authorization mechanisms to verify the identities of users and enforce access policies effectively. It should also incorporate encryption techniques to protect data during transit and at rest, preventing unauthorized access even if the data is compromised.

Ultimately, the goal is to establish a comprehensive access control solution that balances security with usability, enabling secure data sharing and collaboration in cloud environments while mitigating the risks associated with unauthorized access and data exposure.

### **1.1 Problem Statement**

As organizations increasingly rely on cloud storage solutions for data sharing and collaboration, ensuring the security and privacy of shared data becomes paramount. However, traditional access control mechanisms often fall short in adequately safeguarding data from unauthorized access or leakage. The challenge lies in implementing a robust access control system that maintains data confidentiality, integrity, and availability while enabling seamless collaboration among authorized users.

Specifically, the problem entails developing an access control framework for cloud-shared data that addresses various security threats, including insider threats, unauthorized access attempts, and data breaches. This framework should incorporate fine-grained access control policies based on user roles, privileges, and attributes to restrict data access to authorized individuals or groups. Moreover, it should support dynamic access control adjustments to accommodate evolving user permissions and organizational requirements.

- The increase of data breaches in cloud computing puts all users at risk of business problems, highlighting the need to improve security measures.
- To mitigate the growing risks of data threats faced by users in cloud Environment, proactive measures are essential.

### **1.2 Objectives**

To accomplish the project's purpose, the following particular objectives have been established.

- To create an user interface and implementing searchable encryption on the text file uploaded by the user.
- To implement the access control scheme and addressing the problem of revocation by using cryptographic techniques.

### **1.3 Scope of the Project**

The following are the boundaries that have established in the proposed system which defines scope.

- i. In proposed system, we use these two cryptographic techniques that secure the data storage in cloud-based environments, to design a hybrid encryption scheme based on ABE and SSE in such a way that we utilize the best out of both of them.
- ii. The data can be encrypted into cipher text before moving into cloud by using symmetric encryption.
- iii. The revocation mechanism and an access control bounded by policy, that ensures data security.

## CHAPTER 2

### LITERATURE SURVEY

Michalas A and A. Bakas presented a novel technique that enables data owners to link specific policies to specific areas of their cipher texts. The scheme is based on existing symmetric primitives. They combined an in-depth simulation-based security study with an experimental evaluation that demonstrates our scheme's efficacy to demonstrate the accuracy of our methodology [1].

A. Sharma claims in J. Bethencourt study that using a trusted server to store data and handle access control is the only way to enforce such regulations. In their system, a party encrypting data establishes a policy for who can decrypt, and attributes are used to characterize a user's credentials. Consequently, techniques like role-based access control (RBAC) [2]

The safe and effective oblivious storage systems described in the paper by Y. XU and W. Cui concentrate on making use of all available network bandwidth to provide concurrent access through a reliable proxy. However, the performance is limited by the network's latency and bandwidth because the proxy uses the network to carry out a common ORAM protocol. Furthermore, in such proxy environments, several crucial elements like access control and security against active adversaries have not been well investigated [3].

The study by R. Dowsley shows how to create a hybrid encryption scheme that combines SSE and ABE while taking advantage of their respective benefits. Unlike many other methods, we build a revocation process that is based only on SGX's capability and is totally independent of the ABE scheme [4].

The concept of backward privacy for searchable encryption is examined for the first time in the study by R. Bost and B. Minaud. Following the theoretical definitions of several flavors of backward privacy, we propose multiple strategies with varied efficiency trade-offs that achieve both forward and backward privacy. Importantly, our constructs depend on primitives like puncturable encryption schemes and limited pseudo-random functions[5].

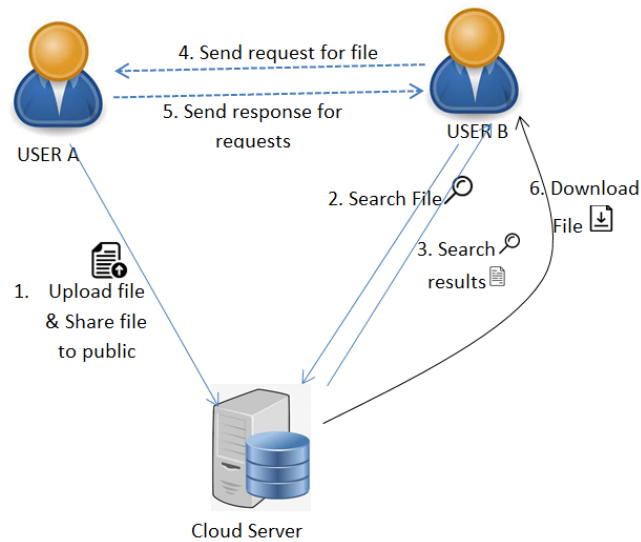
The traditional system says that the cloud service provider will encrypt the user's data and stores in it which is not acceptable by the user because the cloud services can access their data. So to address this problem , they have proposed symmetric searchable encryption where data encrypted before sending into cloud. Here, only user can encrypt and decrypt the document by using encrypted Keyword. The Encrypted Keyword can be generated by the secret key of the user.

The system says that In several distributed systems a user should only be able to access data if a user posses a certain set of credentials or attributes. Currently, the only method for enforcing such policies is to employ a trusted server to store the data and mediate access control. However, if any server storing the data is compromised, then the confidentiality of the data will be compromised. In this paper we present a system for realizing complex access control on encrypted data that we call ciphertext-policy attribute-based encryption.

## CHAPTER 3

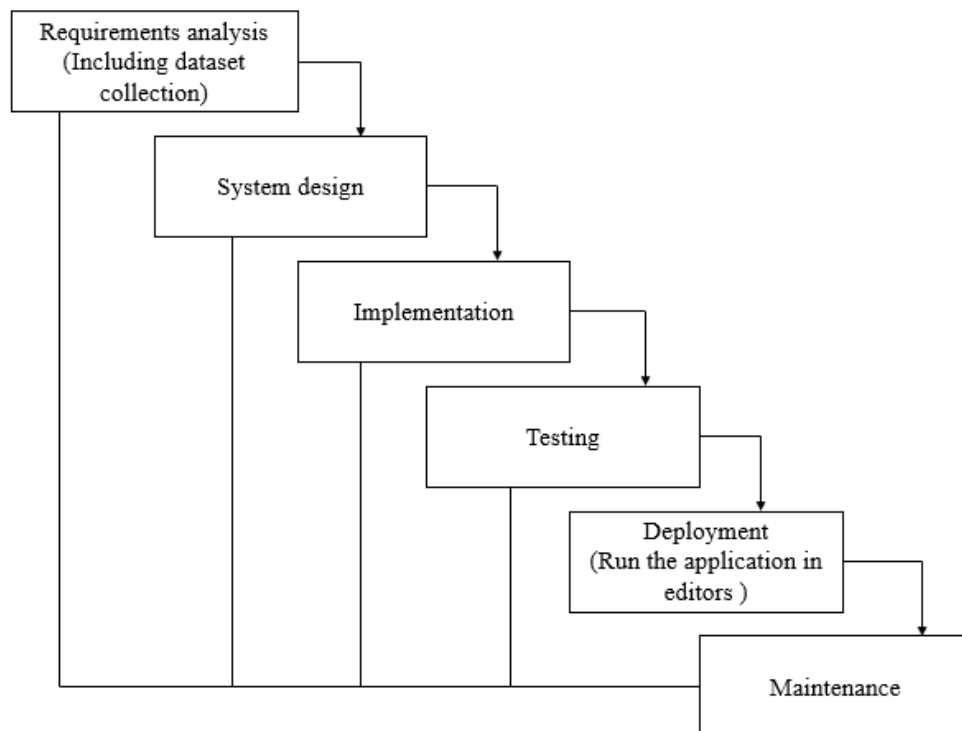
### SYSTEM ANALYSIS & FEASIBILITY STUDY

#### Architecture:



#### SOFTWARE DEVELOPMENT LIFE CYCLE – SDLC

In our project we use waterfall model as our software development cycle because of its step-by-step procedure while implementing.



Waterfall Model

- **Requirement Gathering and analysis** – all possible requirements of the system to be developed are captured in this phase and documented in a requirement specification document.
- **System Design** – the requirement specifications from first phase are studied in this phase and the system design is prepared. This system design helps in specifying hardware and system requirements and helps in defining the overall system architecture.
- **Implementation** – with inputs from the system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality, which is referred to as Unit Testing.
- **Integration and Testing** – All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.
- **Deployment of system** – Once the functional and non-functional testing is done; the product is deployed in the customer environment or released into the market.
- **Maintenance** – There are some issues which come up in the client environment. To fix those issues, patches are released. Also, to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.

## FEASIBILITY STUDY

The feasibility of the project is analysed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.



Three key considerations involved in the feasibility analysis are

- ◆ ECONOMICAL FEASIBILITY
- ◆ TECHNICAL FEASIBILITY
- ◆ SOCIAL FEASIBILITY

**Economic feasibility:**

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus, the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

**Technical feasibility:**

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

**Social feasibility:**

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

## **CHAPTER 4**

### **SYSTEM REQUIREMENTS SPECIFICATION**

#### **4.1 Functional Requirements**

Such requirements describe system behavior under specific conditions and include the product features and functions which web & app developers must add to the solution. Such requirements should be precise both for the development team and stakeholders. The list of examples of functional requirements includes:

- Business Rules
- Transaction corrections, adjustments, and cancellations
- Administrative functions
- Authentication
- Authorization levels
- Audit Tracking
- External Interfaces
- Certification Requirements
- Reporting Requirements
- Historical Data

#### **4.2 Non-Functional Requirements**

Hence few things to be noted before Mobile Application Development are

- Unlike laptops/Desktop the resources available on Mobile devices are very less like processor, speed, screen dimension etc.
- Scalability of device screen is different for different mobiles.
- Network condition may vary.
- Multitasking capability and Memory available.
- Different version of OS and backward compatibility etc.

Moving on let me List out few Key types of NFR that needs to be taken care of

- Performance
- Scalability
- Responsiveness
- Use-ability
- reliability
- Security
- Documentation
- Availability

## **HARDWARE & SOFTWARE REQUIREMENTS**

### **4.3 Hardware System Configurations :-**

- Processor - I3/Intel Processor
- RAM - 4GB (min)
- Hard Disk - 500GB

### **4.4 Software System Configurations :-**

- Operating System : Windows 7/8/10
- Application Server : Tomcat 7.0
- Front End : HTML, JSP
- Scripts : JavaScript.
- Server side Script : Java Server Pages.
- Database : My SQL 6.0
- Database Connectivity : JDBC

## Chapter 5

### System Analysis and Design

#### 5.1 Input Design:

In an information system, input is the raw data that is processed to produce output. During the input design, the developers must consider the input devices such as PC, MICR, OMR, etc.

Therefore, the quality of system input determines the quality of system output. Well-designed input forms and screens have following properties –

- It should serve specific purpose effectively such as storing, recording, and retrieving the information.
- It ensures proper completion with accuracy.
- It should be easy to fill and straightforward.
- It should focus on user's attention, consistency, and simplicity.
- All these objectives are obtained using the knowledge of basic design principles regarding –
  - What are the inputs needed for the system?
  - How end users respond to different elements of forms and screens.

##### 5.1.1 Objectives for Input Design:

The objectives of input design are

- To design data entry and input procedures
- To reduce input volume
- To design source documents for data capture or devise other data capture methods
- To design input data records, data entry screens, user interface screens, etc.
- To use validation checks and develop effective input controls.

## 5.2 Output Design:

The design of output is the most important task of any system. During output design, developers identify the type of outputs needed, and consider the necessary output controls and prototype report layouts.

### 5.2.1 Objectives of Output Design:

The objectives of input design are:

- To develop output design that serves the intended purpose and eliminates the production of unwanted output.
- To develop the output design that meets the end user's requirements.
- To deliver the appropriate quantity of output.
- To form the output in appropriate format and direct it to the right person.
- To make the output available on time for making good decisions.

## 5.3 MODULES:

### MODULES :

#### 5.3.1 User :

- **Register:** user can enter the details and he can register.
- **Login:** user can login with his valid credentials. If user enter invalid credentials then it can be redirect into login page. If user enters valid credentials then it can be redirect into user home.
- **Upload:** Here, User can Upload the files.
- While uploading the file it can be store into encrypted format and generates searchable keywords.
- **View Files :** The user view the uploaded files and share files to other users.
- **Search:** user can search for a file based on keywords. If file has been found send request to file uploaded user

- **View request** : In this user view the request from other users for their file then user can accept/ Reject the request
- **Status** : The user view the file requested status, i.e. pending and Accepted.
- **Download**: user can download the file if his request is accepted. Here Encryption file converted into decryption format (original File) can be Download.

## 5.4 UML DIAGRAMS

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

### GOALS:

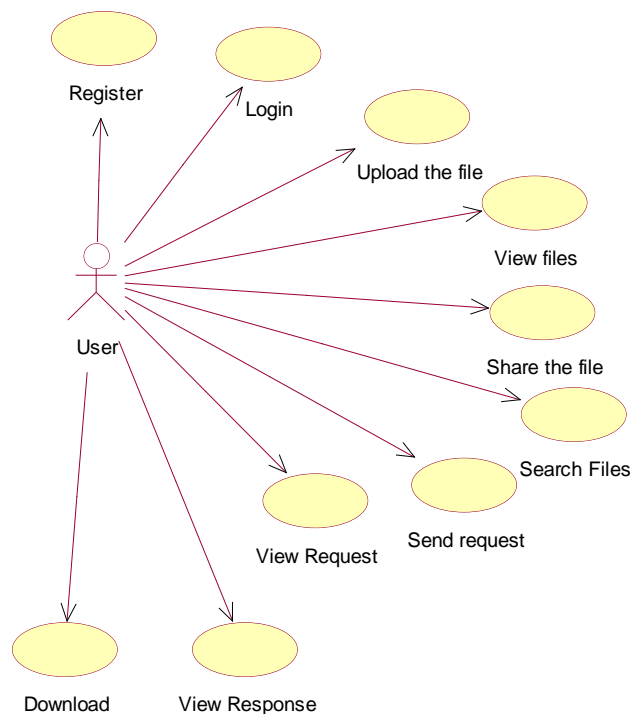
The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.

2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understanding the modeling language.
5. Encourage the growth of OO tools market.
6. Support higher level development concepts such as collaborations, frameworks, patterns and components.
7. Integrate best practices.

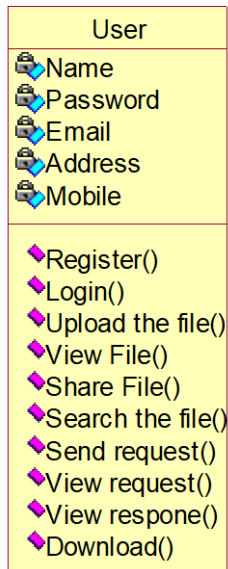
### 5.5 USE CASE DIAGRAM:

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.



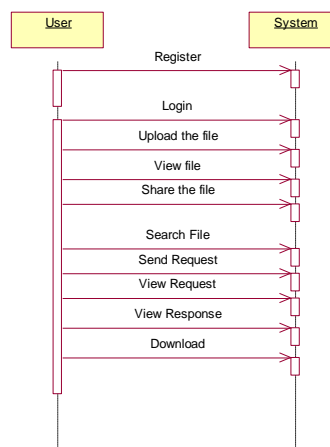
## 5.6 CLASS DIAGRAM:

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.



## 5.7 SEQUENCE DIAGRAM:

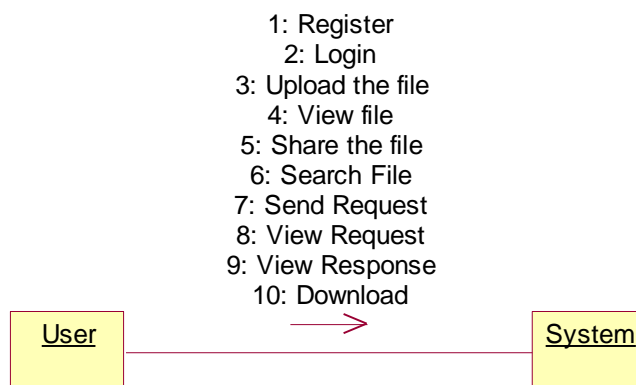
A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.





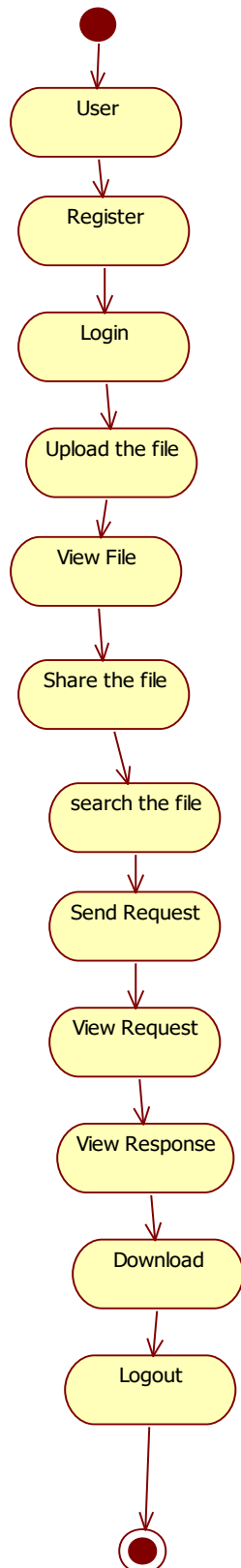
## 5.8 Collaboration Diagram:

In collaboration diagram the method call sequence is indicated by some numbering technique as shown below. The number indicates how the methods are called one after another. We have taken the same order management system to describe the collaboration diagram. The method calls are similar to that of a sequence diagram. But the difference is that the sequence diagram does not describe the object organization whereas the collaboration diagram shows the object organization.



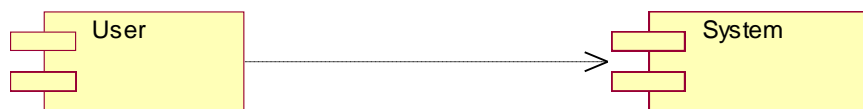
## 5.9 ACTIVITY DIAGRAM:

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.



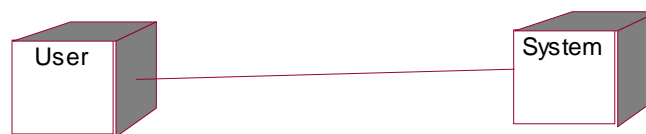
### 5.10 COMPONENT DIAGRAM

Component diagrams are used to describe the physical artifacts of a system. This artifact includes files, executables, libraries etc. So the purpose of this diagram is different, Component diagrams are used during the implementation phase of an application. But it is prepared well in advance to visualize the implementation details. Initially the system is designed using different UML diagrams and then when the artifacts are ready component diagrams are used to get an idea of the implementation.



### 5.11 DEPLOYMENT DIAGRAM:

Deployment diagram represents the deployment view of a system. It is related to the component diagram. Because the components are deployed using the deployment diagrams. A deployment diagram consists of nodes. Nodes are nothing but physical hardware's used to deploy the application.



## CHAPTER 6

### SOFTWARE ENVIRONMENT AND INSTALLATION

#### Software Environment

#### Java Technology

**Java technology is both a programming language and a platform.**

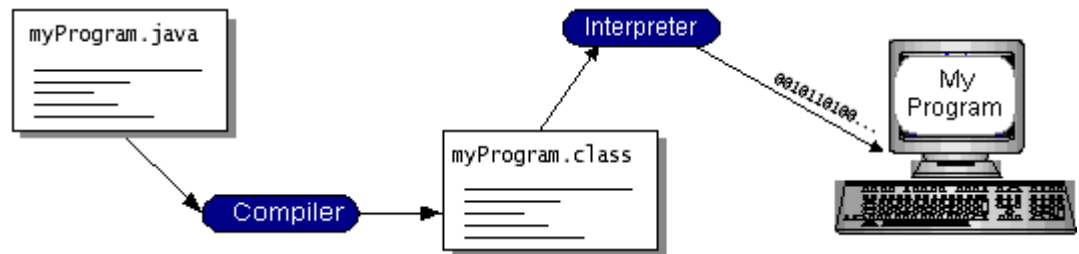
#### The Java Programming Language

The Java programming language is a high-level language that can be characterized by all of the following buzzwords:

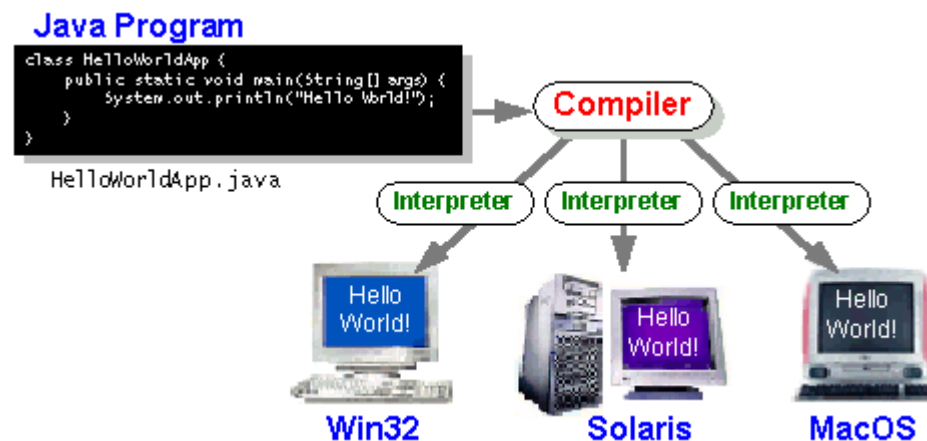
- Simple
- Architecture neutral
- Object oriented
- Portable
- Distributed
- High performance
- Interpreted
- Multithreaded
- Robust
- Dynamic
- Secure

With most programming languages, you either compile or interpret a program so that you can run it on your computer. The Java programming language is unusual in that a program is both compiled and interpreted. With the compiler, first you translate a program into an intermediate language called *Java byte codes* —the platform-independent codes interpreted by the interpreter on the Java platform. The interpreter

parses and runs each Java byte code instruction on the computer. Compilation happens just once; interpretation occurs each time the program is executed. The following figure illustrates how this works.



You can think of Java byte codes as the machine code instructions for the *Java Virtual Machine* (Java VM). Every Java interpreter, whether it's a development tool or a Web browser that can run applets, is an implementation of the Java VM. Java byte codes help make “write once, run anywhere” possible. You can compile your program into byte codes on any platform that has a Java compiler. The byte codes can then be run on any implementation of the Java VM. That means that as long as a computer has a Java VM, the same program written in the Java programming language can run on Windows 2000, a Solaris workstation, or on an iMac.



## The Java Platform

A *platform* is the hardware or software environment in which a program runs. We've

already mentioned some of the most popular platforms like Windows 2000, Linux, Solaris, and MacOS. Most platforms can be described as a combination of the operating system and hardware. The Java platform differs from most other platforms in that it's a software-only platform that runs on top of other hardware-based platforms.

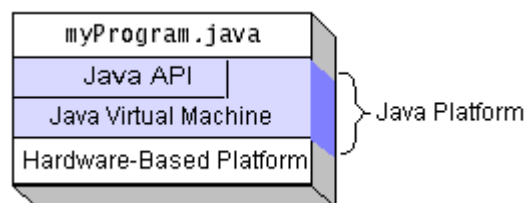
The Java platform has two components:

- *The Java Virtual Machine (Java VM)*
- *The Java Application Programming Interface (Java API)*

You've already been introduced to the Java VM. It's the base for the Java platform and is ported onto various hardware-based platforms.

The Java API is a large collection of ready-made software components that provide many useful capabilities, such as graphical user interface (GUI) widgets. The Java API is grouped into libraries of related classes and interfaces; these libraries are known as *packages*. The next section, What Can Java Technology Do? Highlights what functionality some of the packages in the Java API provide.

The following figure depicts a program that's running on the Java platform. As the figure shows, the Java API and the virtual machine insulate the program from the hardware.



Native code is code that after you compile it, the compiled code runs on a specific hardware platform. As a platform-independent environment, the Java platform can be a bit slower than native code. However, smart compilers, well-tuned interpreters, and just-in-time byte code compilers can bring performance close to that of native code without threatening portability.

### **What Can Java Technology Do?**

The most common types of programs written in the Java programming language are

*applets* and *applications*. If you've surfed the Web, you're probably already familiar with applets. An applet is a program that adheres to certain conventions that allow it to run within a Java-enabled browser.

However, the Java programming language is not just for writing cute, entertaining applets for the Web. The general-purpose, high-level Java programming language is also a powerful software platform. Using the generous API, you can write many types of programs.

An application is a standalone program that runs directly on the Java platform. A special kind of application known as a *server* serves and supports clients on a network. Examples of servers are Web servers, proxy servers, mail servers, and print servers. Another specialized program is a *servlet*. A servlet can almost be thought of as an applet that runs on the server side. Java Servlets are a popular choice for building interactive web applications, replacing the use of CGI scripts. Servlets are similar to applets in that they are runtime extensions of applications. Instead of working in browsers, though, servlets run within Java Web servers, configuring or tailoring the server.

How does the API support all these kinds of programs? It does so with packages of software components that provides a wide range of functionality. Every full implementation of the Java platform gives you the following features:

**The essentials:** Objects, strings, threads, numbers, input and output, data structures, system properties, date and time, and so on.

**Applets:** The set of conventions used by applets.

**Networking:** URLs, TCP (Transmission Control Protocol), UDP (User Datagram Protocol) sockets, and IP (Internet Protocol) addresses.

**Internationalization:** Help for writing programs that can be localized for users worldwide. Programs can automatically adapt to specific locales and be displayed in the appropriate language.

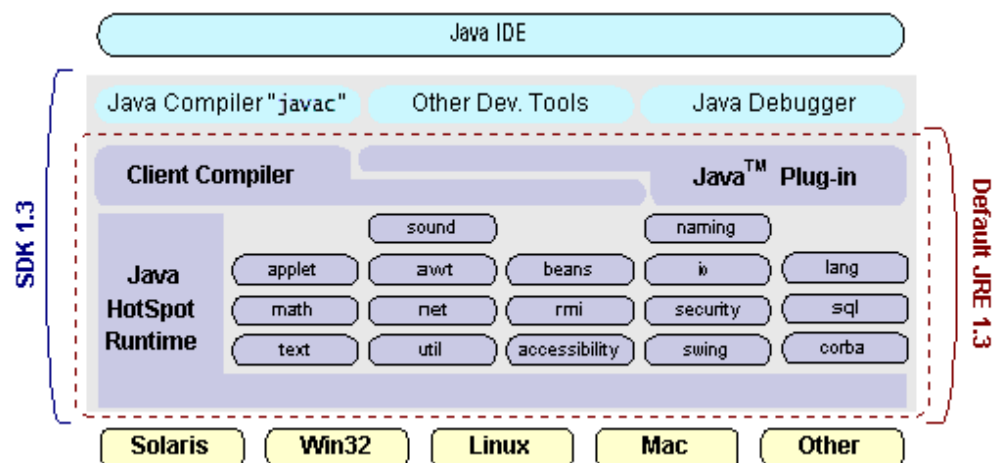
**Security:** Both low level and high level, including electronic signatures, public and private key management, access control, and certificates.

**Software components:** Known as JavaBeans™, can plug into existing component architectures.

**Object serialization:** Allows lightweight persistence and communication via Remote Method Invocation (RMI).

**Java Database Connectivity (JDBC™):** Provides uniform access to a wide range of relational databases.

The Java platform also has APIs for 2D and 3D graphics, accessibility, servers, collaboration, telephony, speech, animation, and more. The following figure depicts what is included in the Java 2 SDK.



### How Will Java Technology Change My Life?

We can't promise you fame, fortune, or even a job if you learn the Java programming language. Still, it is likely to make your programs better and requires less effort than other languages. We believe that Java technology will help you do the following:

**Get started quickly:** Although the Java programming language is a powerful object-oriented language, it's easy to learn, especially for programmers already familiar with C or C++.

**Write less code:** Comparisons of program metrics (class counts, method counts, and so on) suggest that a program written in the Java programming language can be four times smaller than the same program in C++.



**Write better code:** The Java programming language encourages good coding practices, and its garbage collection helps you avoid memory leaks. Its object orientation, its JavaBeans component architecture, and its wide-ranging, easily extendible API let you reuse other people's tested code and introduce fewer bugs.

**Develop programs more quickly:** Your development time may be as much as twice as fast versus writing the same program in C++. Why? You write fewer lines of code and it is a simpler programming language than C++.

**Avoid platform dependencies with 100% Pure Java:** You can keep your program portable by avoiding the use of libraries written in other languages. The 100% Pure Java™ Product Certification Program has a repository of historical process manuals, white papers, brochures, and similar materials online.

**Write once, run anywhere:** Because 100% Pure Java programs are compiled into machine-independent byte codes, they run consistently on any Java platform.

**Distribute software more easily:** You can upgrade applets easily from a central server. Applets take advantage of the feature of allowing new classes to be loaded "on the fly," without recompiling the entire program.

## **ODBC**

Microsoft Open Database Connectivity (ODBC) is a standard programming interface for application developers and database systems providers. Before ODBC became a *de facto* standard for Windows programs to interface with database systems, programmers had to use proprietary languages for each database they wanted to connect to. Now, ODBC has made the choice of the database system almost irrelevant from a coding perspective, which is as it should be. Application developers have much more important things to worry about than the syntax that is needed to port their program from one database to another when business needs suddenly change.

Through the ODBC Administrator in Control Panel, you can specify the particular database that is associated with a data source that an ODBC application program is written to use. Think of an ODBC data source as a door with a name on it. Each door will lead you to a particular database. For example, the data source named Sales Figures might be a SQL Server database, whereas the Accounts Payable data source could refer

to an Access database. The physical database referred to by a data source can reside anywhere on the LAN.

The ODBC system files are not installed on your system by Windows 95. Rather, they are installed when you setup a separate database application, such as SQL Server Client or Visual Basic 4.0. When the ODBC icon is installed in Control Panel, it uses a file called ODBCINST.DLL. It is also possible to administer your ODBC data sources through a stand-alone program called ODBCADM.EXE. There is a 16-bit and a 32-bit version of this program and each maintains a separate list of ODBC data sources. From a programming perspective, the beauty of ODBC is that the application can be written to use the same set of function calls to interface with any data source, regardless of the database vendor. The source code of the application doesn't change whether it talks to Oracle or SQL Server. We only mention these two as an example. There are ODBC drivers available for several dozen popular database systems. Even Excel spreadsheets and plain text files can be turned into data sources. The operating system uses the Registry information written by ODBC Administrator to determine which low-level ODBC drivers are needed to talk to the data source (such as the interface to Oracle or SQL Server). The loading of the ODBC drivers is transparent to the ODBC application program. In a client/server environment, the ODBC API even handles many of the network issues for the application programmer.

The advantages of this scheme are so numerous that you are probably thinking there must be some catch. The only disadvantage of ODBC is that it isn't as efficient as talking directly to the native database interface. ODBC has had many detractors make the charge that it is too slow. Microsoft has always claimed that the critical factor in performance is the quality of the driver software that is used. In our humble opinion, this is true. The availability of good ODBC drivers has improved a great deal recently. And anyway, the criticism about performance is somewhat analogous to those who said that compilers would never match the speed of pure assembly language. Maybe not, but the compiler (or ODBC) gives you the opportunity to write cleaner programs, which means you finish sooner. Meanwhile, computers get faster every year.

## JDBC

In an effort to set an independent database standard API for Java; Sun Microsystems developed Java Database Connectivity, or JDBC. JDBC offers a generic SQL database access mechanism that provides a consistent interface to a variety of RDBMSs. This consistent interface is achieved through the use of “plug-in” database connectivity modules, or *drivers*. If a database vendor wishes to have JDBC support, he or she must provide the driver for each platform that the database and Java run on.

To gain a wider acceptance of JDBC, Sun based JDBC’s framework on ODBC. As you discovered earlier in this chapter, ODBC has widespread support on a variety of platforms. Basing JDBC on ODBC will allow vendors to bring JDBC drivers to market much faster than developing a completely new connectivity solution.

JDBC was announced in March of 1996. It was released for a 90 day public review that ended June 8, 1996. Because of user input, the final JDBC v1.0 specification was released soon after.

The remainder of this section will cover enough information about JDBC for you to know what it is about and how to use it effectively. This is by no means a complete overview of JDBC. That would fill an entire book.

### JDBC Goals

Few software packages are designed without goals in mind. JDBC is one that, because of its many goals, drove the development of the API. These goals, in conjunction with early reviewer feedback, have finalized the JDBC class library into a solid framework for building database applications in Java.

The goals that were set for JDBC are important. They will give you some insight as to why certain classes and functionalities behave the way they do. The eight design goals for JDBC are as follows:

### SQL Level API

The designers felt that their main goal was to define a SQL interface for Java. Although not the lowest database interface level possible, it is at a low enough level for higher-

level tools and APIs to be created. Conversely, it is at a high enough level for application programmers to use it confidently. Attaining this goal allows for future tool vendors to “generate” JDBC code and to hide many of JDBC’s complexities from the end user.

### **1. SQL Conformance**

SQL syntax varies as you move from database vendor to database vendor. In an effort to support a wide variety of vendors, JDBC will allow any query statement to be passed through it to the underlying database driver. This allows the connectivity module to handle non-standard functionality in a manner that is suitable for its users.

### **2. JDBC must be implemental on top of common database interfaces**

The JDBC SQL API must “sit” on top of other common SQL level APIs. This goal allows JDBC to use existing ODBC level drivers by the use of a software interface. This interface would translate JDBC calls to ODBC and vice versa.

### **3. Provide a Java interface that is consistent with the rest of the Java system**

Because of Java’s acceptance in the user community thus far, the designers feel that they should not stray from the current design of the core Java system.

### **4. Keep it simple**

This goal probably appears in all software design goal listings. JDBC is no exception. Sun felt that the design of JDBC should be very simple, allowing for only one method of completing a task per mechanism. Allowing duplicate functionality only serves to confuse the users of the API.

### **5. Use strong, static typing wherever possible**

Strong typing allows for more error checking to be done at compile time; also, less error appear at runtime.

### **6. Keep the common cases simple**

Because more often than not, the usual SQL calls used by the programmer are simple SELECT's, INSERT's, DELETE's and UPDATE's, these queries should be simple to perform with JDBC. However, more complex SQL statements should also be possible.

Finally we decided to proceed the implementation using Java Networking.

And for dynamically updating the cache table we go for MS Access database.

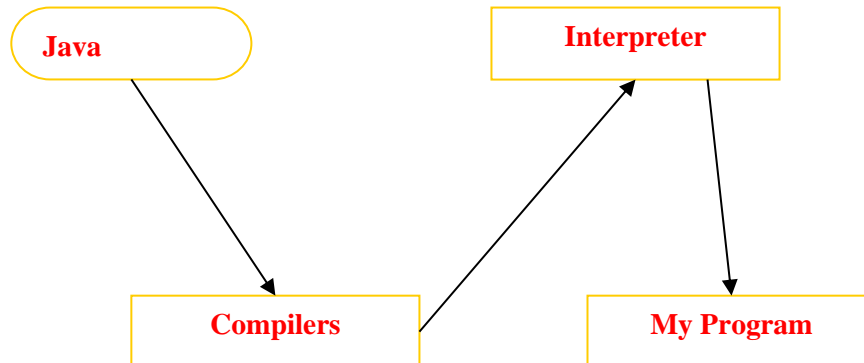
Java has two things: a programming language and a platform.

Java is a high-level programming language that is all of the following

- Simple
- Architecture-neutral
- Object-oriented
- Portable
- Distributed
- High-performance
- Interpreted
- Multithreaded
- Robust
- Dynamic
- Secure

Java is also unusual in that each Java program is both compiled and interpreted. With a compile you translate a Java program into an intermediate language called Java byte codes the platform-independent code instruction is passed and run on the computer.

Compilation happens just once; interpretation occurs each time the program is executed. The figure illustrates how this works.



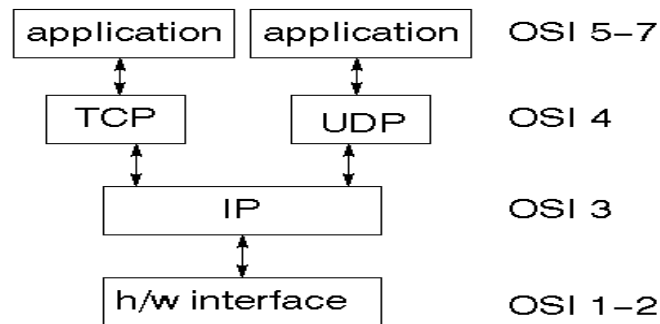
You can think of Java byte codes as the machine code instructions for the Java Virtual Machine (Java VM). Every Java interpreter, whether it's a Java development tool or a Web browser that can run Java applets, is an implementation of the Java VM. The Java VM can also be implemented in hardware.

Java byte codes help make “write once, run anywhere” possible. You can compile your Java program into byte codes on my platform that has a Java compiler. The byte codes can then be run any implementation of the Java VM. For example, the same Java program can run Windows NT, Solaris, and Macintosh.

## **Networking**

### **TCP/IP stack**

The TCP/IP stack is shorter than the OSI one:



TCP is a connection-oriented protocol; UDP (User Datagram Protocol) is a connectionless protocol.

### IP datagram's

The IP layer provides a connectionless and unreliable delivery system. It considers each datagram independently of the others. Any association between datagram must be supplied by the higher layers. The IP layer supplies a checksum that includes its own header. The header includes the source and destination addresses. The IP layer handles routing through an Internet. It is also responsible for breaking up large datagram into smaller ones for transmission and reassembling them at the other end.

### UDP

UDP is also connectionless and unreliable. What it adds to IP is a checksum for the contents of the datagram and port numbers. These are used to give a client/server model - see later.

### TCP

TCP supplies logic to give a reliable connection-oriented protocol above IP. It provides a virtual circuit that two processes can use to communicate.

### Internet addresses

In order to use a service, you must be able to find it. The Internet uses an address scheme for machines so that they can be located. The address is a 32 bit integer which gives the IP address. This encodes a network ID and more addressing. The network ID falls into various classes according to the size of the network address.

## Network address

Class A uses 8 bits for the network address with 24 bits left over for other addressing. Class B uses 16 bit network addressing. Class C uses 24 bit network addressing and class D uses all 32.

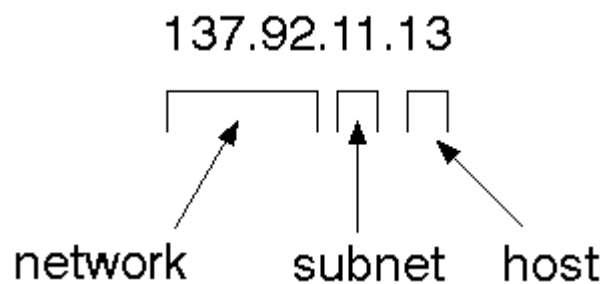
## Subnet address

Internally, the UNIX network is divided into sub networks. Building 11 is currently on one sub network and uses 10-bit addressing, allowing 1024 different hosts.

## Host address

8 bits are finally used for host addresses within our subnet. This places a limit of 256 machines that can be on the subnet.

## Total address



The 32 bit address is usually written as 4 integers separated by dots.

## Port addresses

A service exists on a host, and is identified by its port. This is a 16 bit number. To send a message to a server, you send it to the port for that service of the host that it is running on. This is not location transparency! Certain of these ports are "well known".

## Sockets

A socket is a data structure maintained by the system to handle network connections. A socket is created using the call `socket`. It returns an integer that is like a file descriptor. In fact, under Windows, this handle can be used with Read File and Write File functions.



```
#include <sys/types.h>
```

```
#include <sys/socket.h>
```

```
int socket(int family, int type, int protocol);
```

Here "family" will be AF\_INET for IP communications, protocol will be zero, and type will depend on whether TCP or UDP is used. Two processes wishing to communicate over a network create a socket each. These are similar to two ends of a pipe - but the actual pipe does not yet exist.

## **JFree Chart**

JFreeChart is a free 100% Java chart library that makes it easy for developers to display professional quality charts in their applications. JFreeChart's extensive feature set includes:

A consistent and well-documented API, supporting a wide range of chart types;

A flexible design that is easy to extend, and targets both server-side and client-side applications;

Support for many output types, including Swing components, image files (including PNG and JPEG), and vector graphics file formats (including PDF, EPS and SVG);

JFreeChart is "open source" or, more specifically, free software. It is distributed under the terms of the GNU Lesser General Public Licence (LGPL), which permits use in proprietary applications.

### **1. Map Visualizations**

Charts showing values that relate to geographical areas. Some examples include: (a) population density in each state of the United States, (b) income per capita for each country in Europe, (c) life expectancy in each country of the world. The tasks in this project include:

Sourcing freely redistributable vector outlines for the countries of the world, states/provinces in particular countries (USA in particular, but also other areas);

Creating an appropriate dataset interface (plus default implementation), a rendered, and

integrating this with the existing XYPlot class in JFreeChart;

Testing, documenting, testing some more, documenting some more.

## **2. Time Series Chart Interactivity**

Implement a new (to JFreeChart) feature for interactive time series charts --- to display a separate control that shows a small version of ALL the time series data, with a sliding "view" rectangle that allows you to select the subset of the time series data to display in the main chart.

## **3. Dashboards**

There is currently a lot of interest in dashboard displays. Create a flexible dashboard mechanism that supports a subset of JFreeChart chart types (dials, pies, thermometers, bars, and lines/time series) that can be delivered easily via both Java Web Start and an applet.

## **4. Property Editors**

The property editor mechanism in JFreeChart only handles a small subset of the properties that can be set for charts. Extend (or reimplement) this mechanism to provide greater end-user control over the appearance of the charts.

# **SOFTWARE INSTALLATION FOR JAVA PROJECTS**

This Java Development Kit(JDK) allows you to code and run Java programs. It's possible that you install multiple JDK versions on the same PC. But Its recommended that you install only latest version.

## **How to install Java for Windows**

Following are the steps for JDK 8 free download for 32 bit or JDK 8 download 64 bit and installation

**Step 1)** Go to [link](#). Click on JDK Download for Java

## Java SE 8

Java SE 8u271 is the latest release for the Java SE 8 Platform.

- [Documentation](#)
- [Installation Instructions](#)
- [Release Notes](#)
- [Oracle License](#)
  - [Binary License](#)
  - [Documentation License](#)
  - [BSD License](#)
- [Java SE Licensing Information User Manual](#)
  - [Includes Third Party Licenses](#)
- [Certified System Configurations](#)
- [Readme Files](#)
  - [JDK ReadMe](#)
  - [JRE ReadMe](#)

### Oracle JDK



JDK Download



Server JRE Download



[JRE Download](#)



Documentation Download



Demos and Samples Download



**Step 2)** Next,

1. Accept License Agreement
2. Download Java 8 JDK for your version 32 bit or JDK 8 download for windows 10 64 bit.

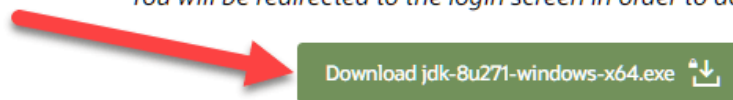
Solaris SPARC 64-bit	88.75 MB	<a href="#">jdk-8u271-solaris-sparcv9.tar.gz</a>
Solaris x64 (SVR4 package)	134.42 MB	<a href="#">jdk-8u271-solaris-x64.tar.Z</a>
Solaris x64	92.52 MB	<a href="#">jdk-8u271-solaris-x64.tar.gz</a>
Windows x86	154.48 MB	<a href="#">jdk-8u271-windows-i586.exe</a>
Windows x64	166.79 MB	<a href="#">jdk-8u271-windows-x64.exe</a>

**Step 3)** 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 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.

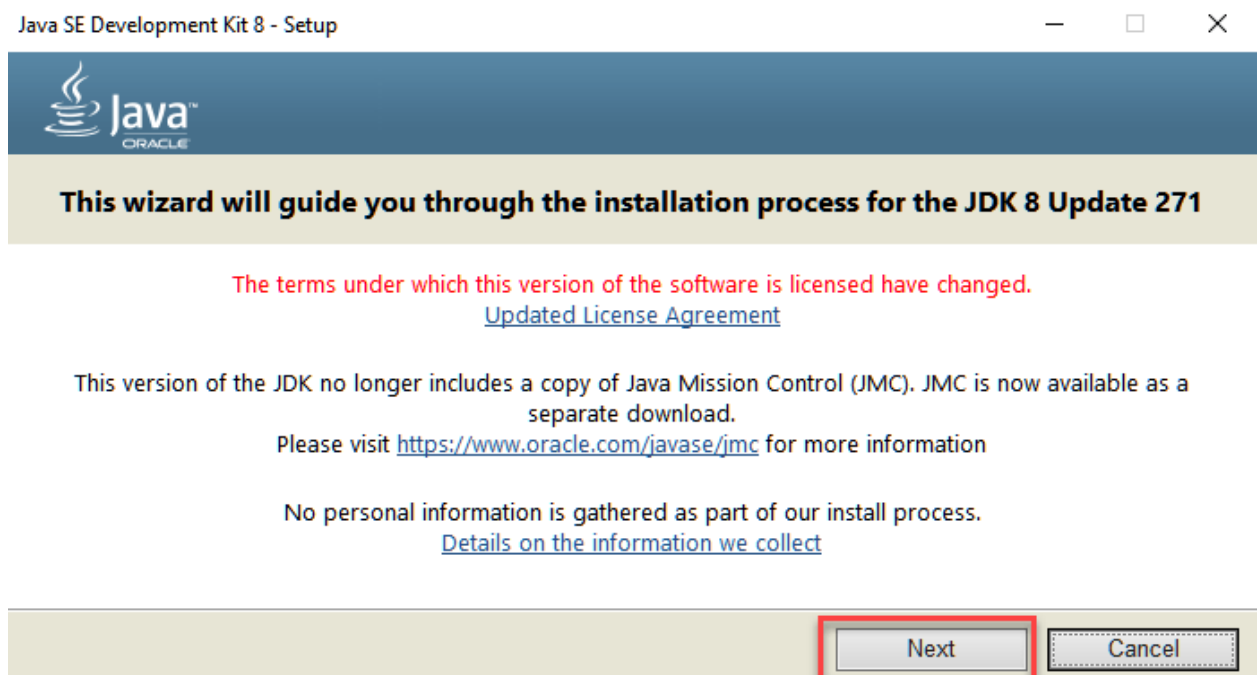
You must accept the Oracle Technology Network License Agreement for Oracle Java SE to download this software. ✕

☒ I reviewed and accept the Oracle Technology Network License Agreement for Oracle Java SE

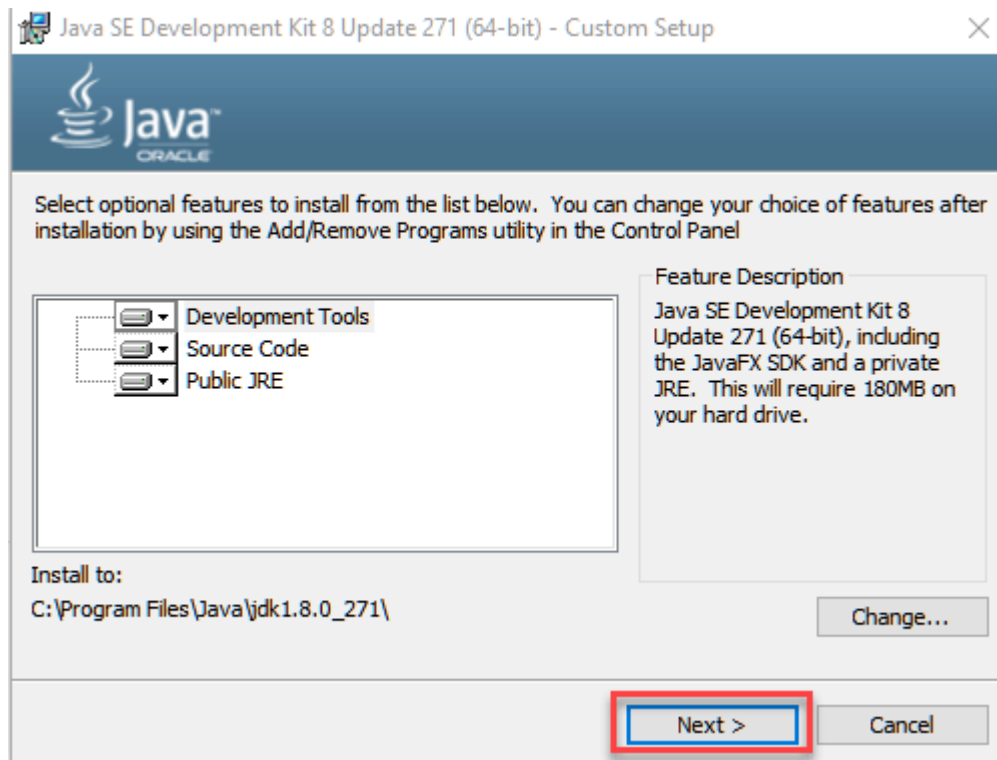
*You will be redirected to the login screen in order to download the file.*



**Step 4)** Once the Java JDK 8 download is complete, run the exe for install JDK. Click Next



**Step 5)** Select the PATH to install Java in Windows and click next.



**Step 6)** Once you install Java in windows, click Close



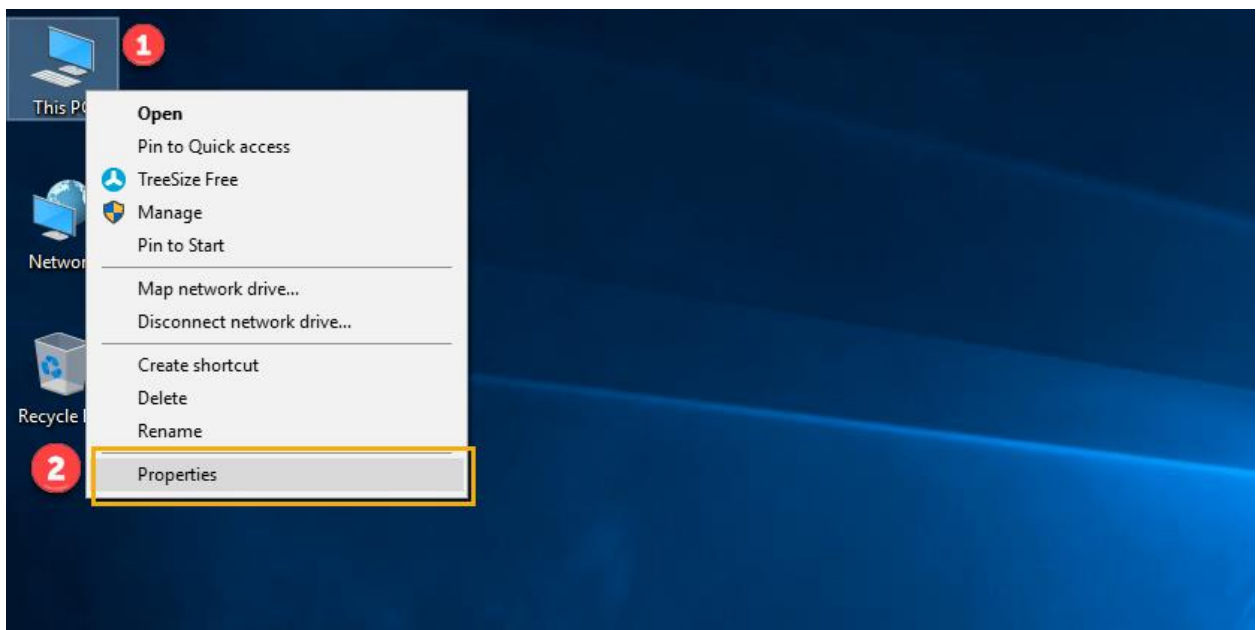
## How to set Environment Variables in Java: Path and Class path

The PATH variable gives the location of executable like javac, java etc. It is possible to run a program without specifying the PATH but you will need to give full path of executable like **C:\Program Files\Java\jdk-13.0.1\bin\javac A.java** instead of simple **javac A.java**

The CLASSPATH variable gives location of the Library Files.

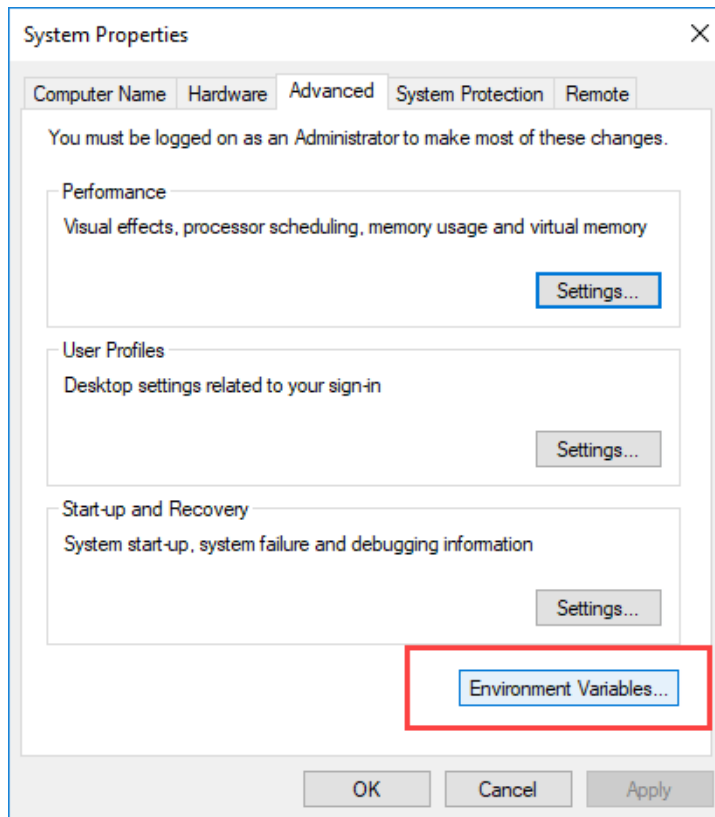
Let's look into the steps to set the PATH and CLASSPATH

**Step 1)** Right Click on the My Computer and Select the properties

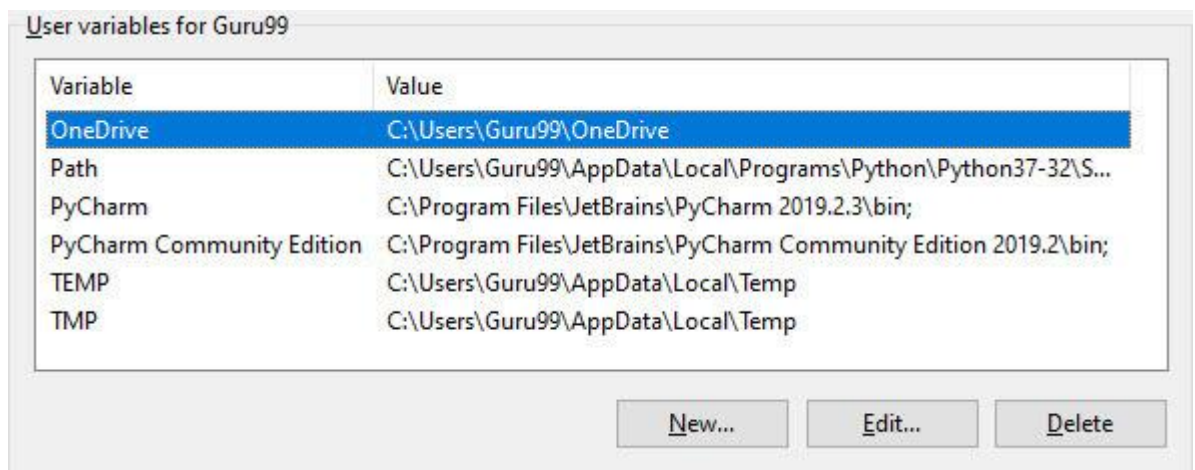


**Step 2)** Click on advanced system settings

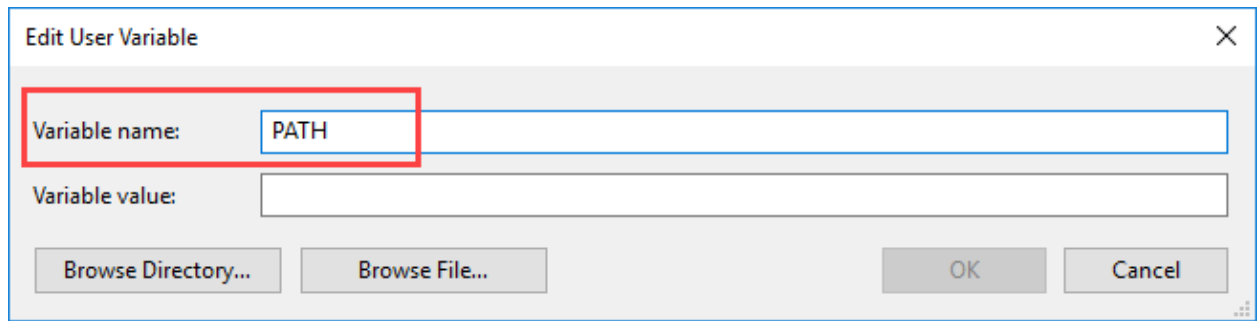
**Step 3)** Click on Environment Variables



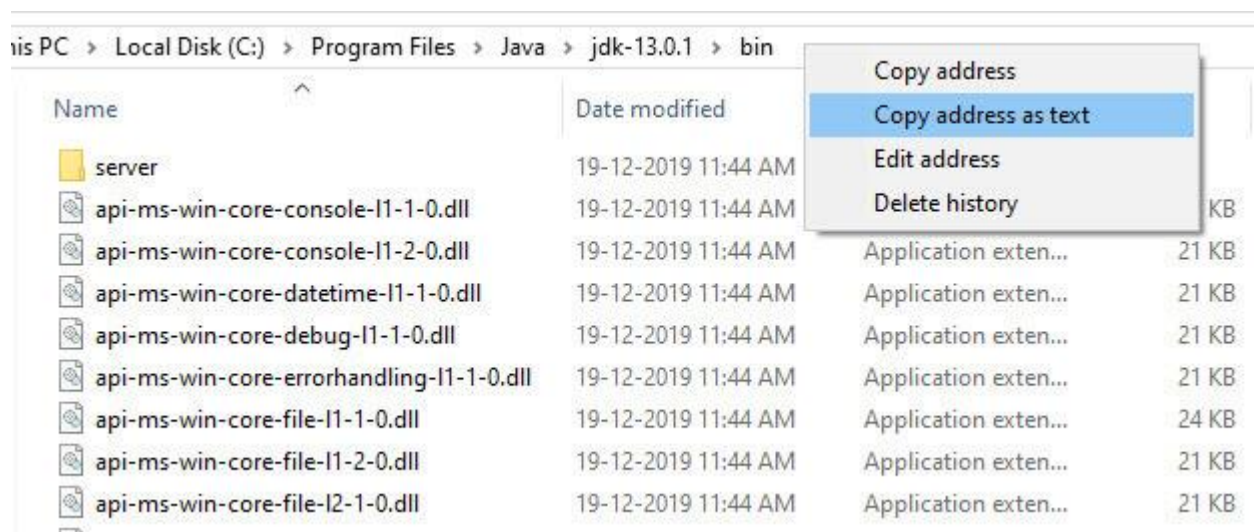
**Step 4)** Click on new Button of User variables



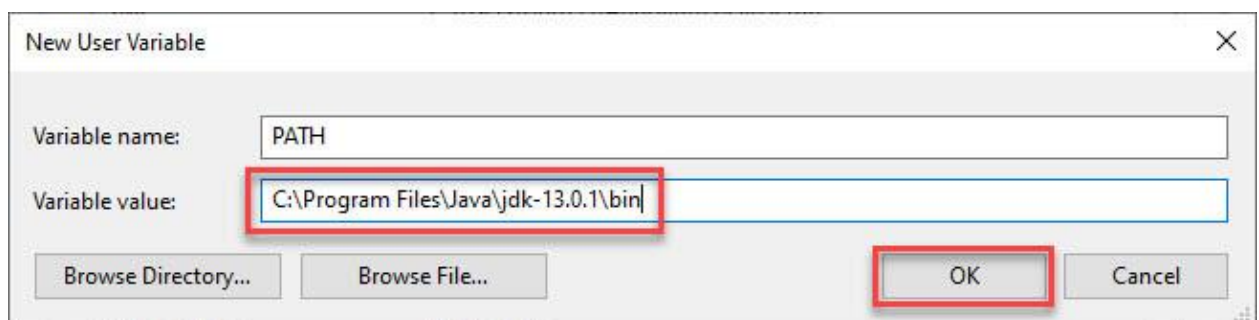
**Step 5)** Type PATH in the Variable name.



**Step 6)** Copy the path of bin folder which is installed in JDK folder.



**Step 7)** Paste Path of bin folder in Variable value and click on OK Button.



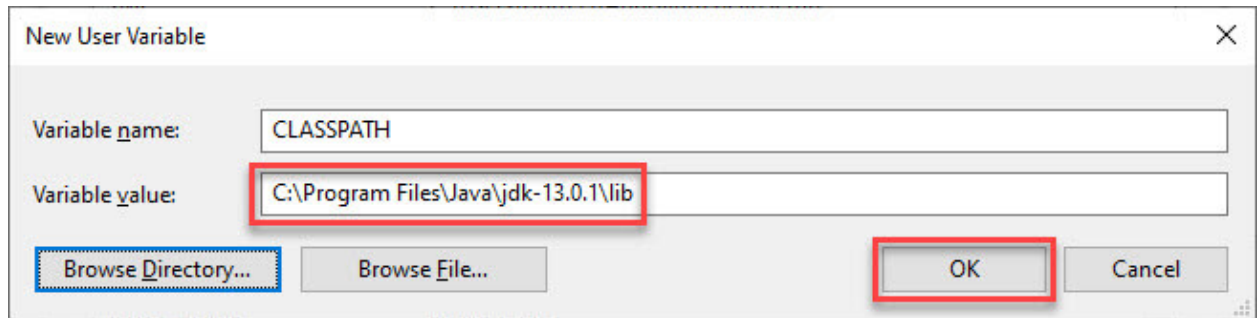
**Note:** In case you already have a PATH variable created in your PC, edit the PATH variable to

`PATH = <JDK installation directory>\bin;%PATH%;`

Here, %PATH% appends the existing path variable to our new value



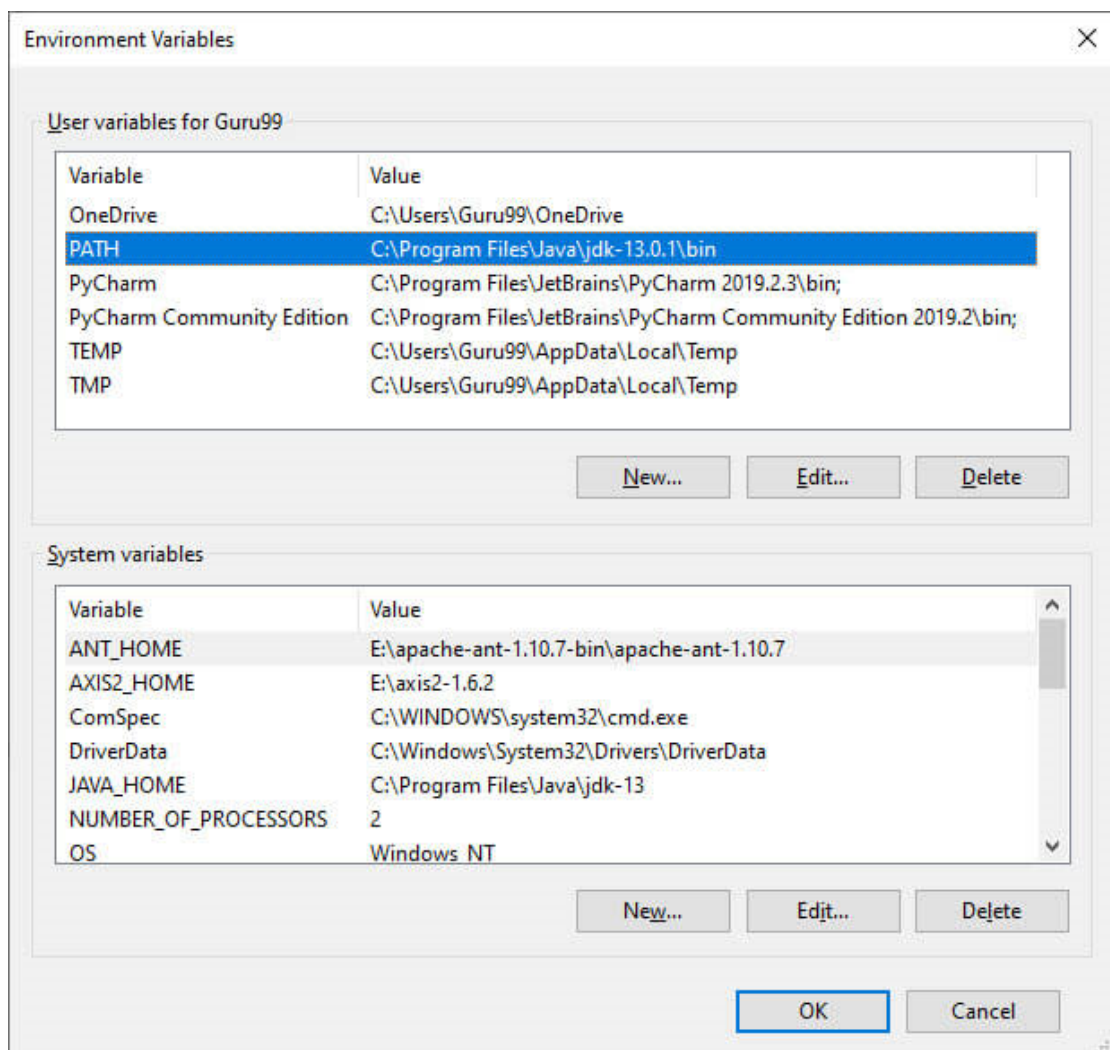
**Step 8)** You can follow a similar process to set CLASSPATH.



**Note:** In case you java installation does not work after installation, change classpath to

CLASSPATH = <JDK installation directory>\lib\tools.jar;

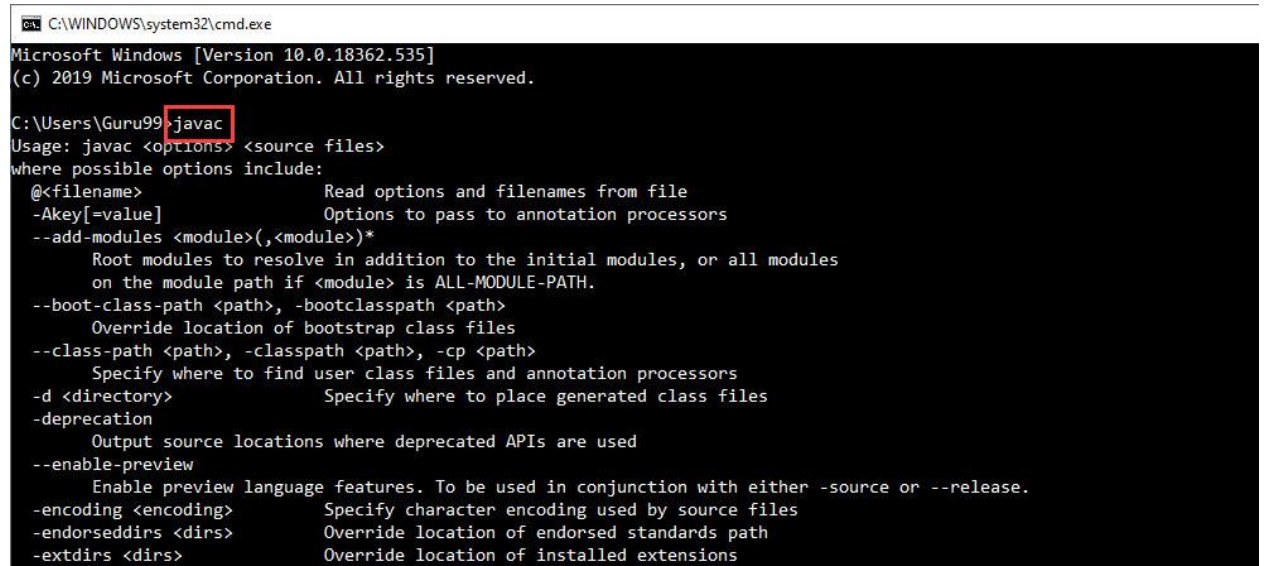
**Step 9)** Click on OK



button

**Step 10)** Go to command prompt and type javac commands.

If you see a screen like below, Java is installed.



```

C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.18362.535]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\Guru99>javac
Usage: javac <options> <source files>
where possible options include:
  @<filename>                Read options and filenames from file
  -Akey[=value]              Options to pass to annotation processors
  --add-modules <module>(<module>)*
                             Root modules to resolve in addition to the initial modules, or all modules
                             on the module path if <module> is ALL-MODULE-PATH.
  --boot-class-path <path>, -bootclasspath <path>
                             Override location of bootstrap class files
  --class-path <path>, -classpath <path>, -cp <path>
                             Specify where to find user class files and annotation processors
  -d <directory>             Specify where to place generated class files
  -deprecation               Output source locations where deprecated APIs are used
  --enable-preview           Enable preview language features. To be used in conjunction with either -source or --release.
  -encoding <encoding>      Specify character encoding used by source files
  -endorseddirs <dirs>      Override location of endorsed standards path
  -extdirs <dirs>           Override location of installed extensions
  
```

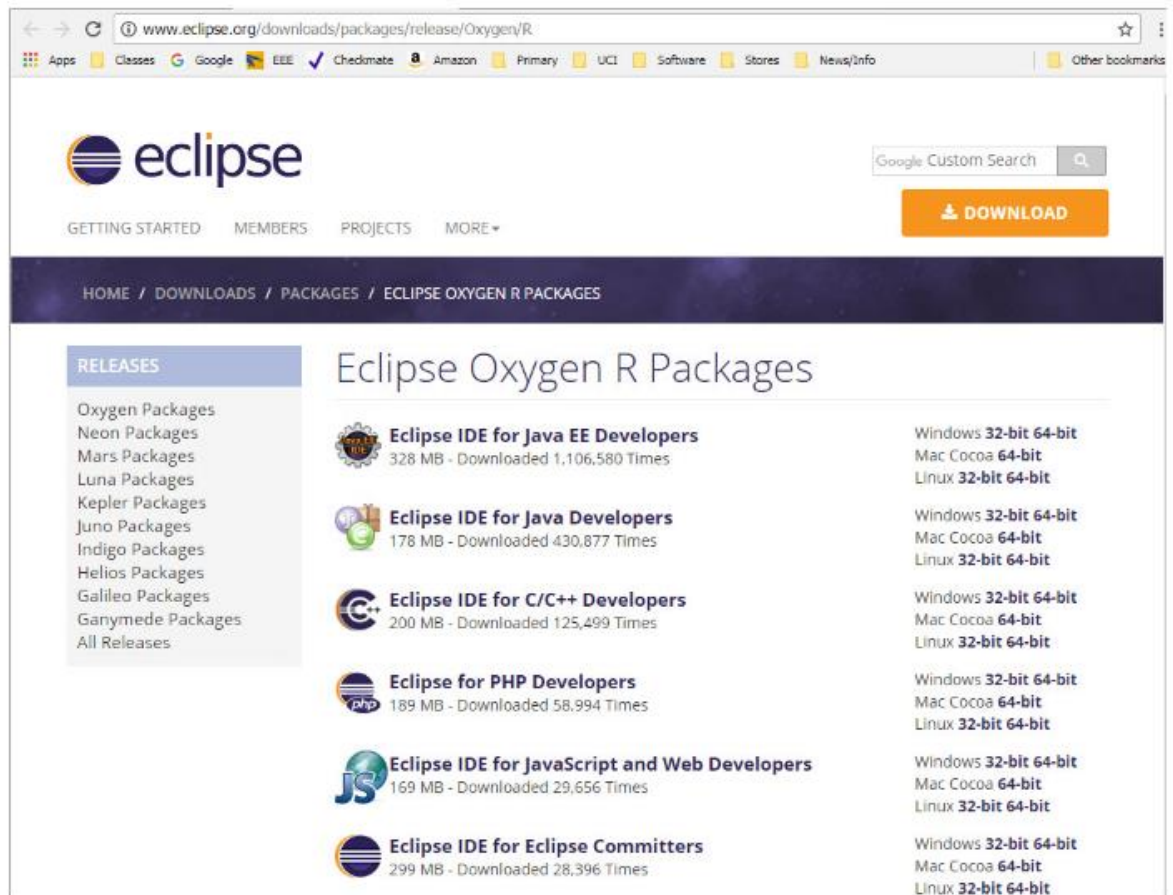
### Eclipse: (Oxygen)

The Eclipse download requires about 300 MB of disk space; keep it on your machine, in case you need to re-install Eclipse. When installed, Eclipse requires an additional 330 MB of disk space.

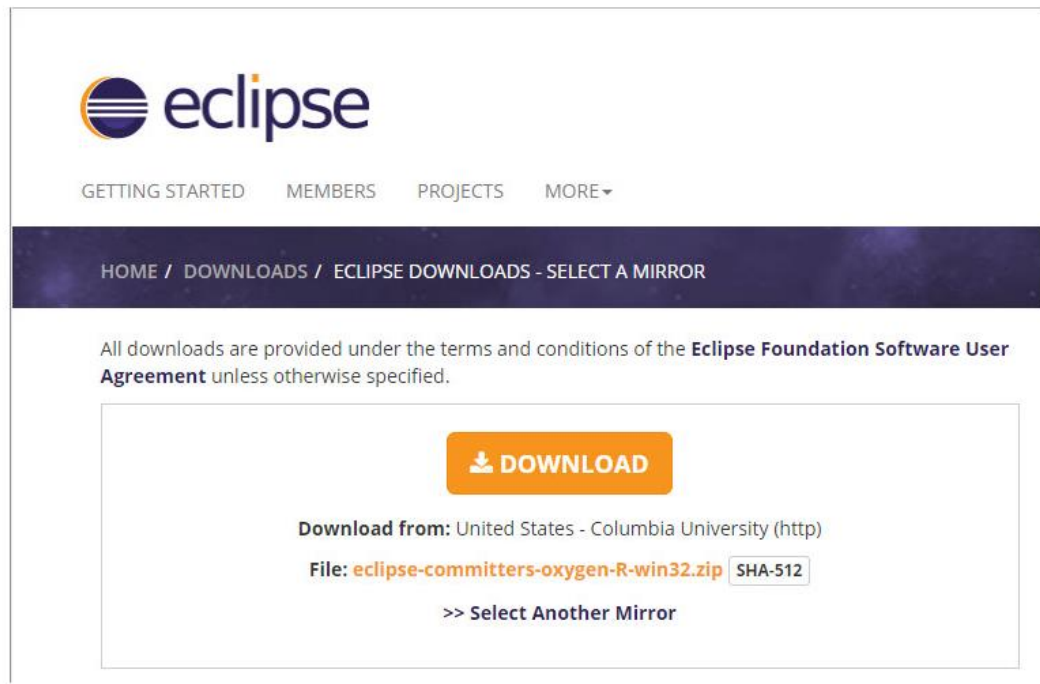
### Downloading

#### 1.Click Eclipse

The following page will appear in your browser. In this handout we will download **Eclipse IDE for Eclipse Committers** for Windows 32 Bit; if your computer uses Windows, continue below; otherwise choose either **Mac Cocoa** or **Linux** instead.

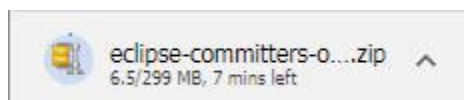


1. It is critical that Java, Python, and Eclipse are either all 32 Bit or are all 64 Bit (and only if your Machine/OS supports 64 Bit): I think it easiest to use 32 Bit for everything.
2. Click the **32-Bit** (after Windows) to the right of the **Eclipse IDE for Eclipse Committers**.
3. You will see the following page (don't worry about the name of the institution underneath the orange **DOWNLOAD** button).

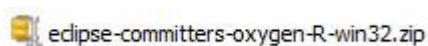


Click the orange **DOWNLOAD** button. The site named here, in orange to the right of the button: **United States - Columbia University (http)** is the random one chosen by the download page this time; yours may differ.

This file should start downloading in your standard download folder, while showing a splash screen about donating to Eclipse. This file is about 300 Mb so it might take a while to download fully if you are on a slow internet connection (it took me about 5 minutes over a cable modem). Don't worry about the exact time as long as the download continues to make steady progress. In Chrome progress is shown on the bottom-left of the window, via the icon



The file should appear as



Terminate the tab browsing this webpage.

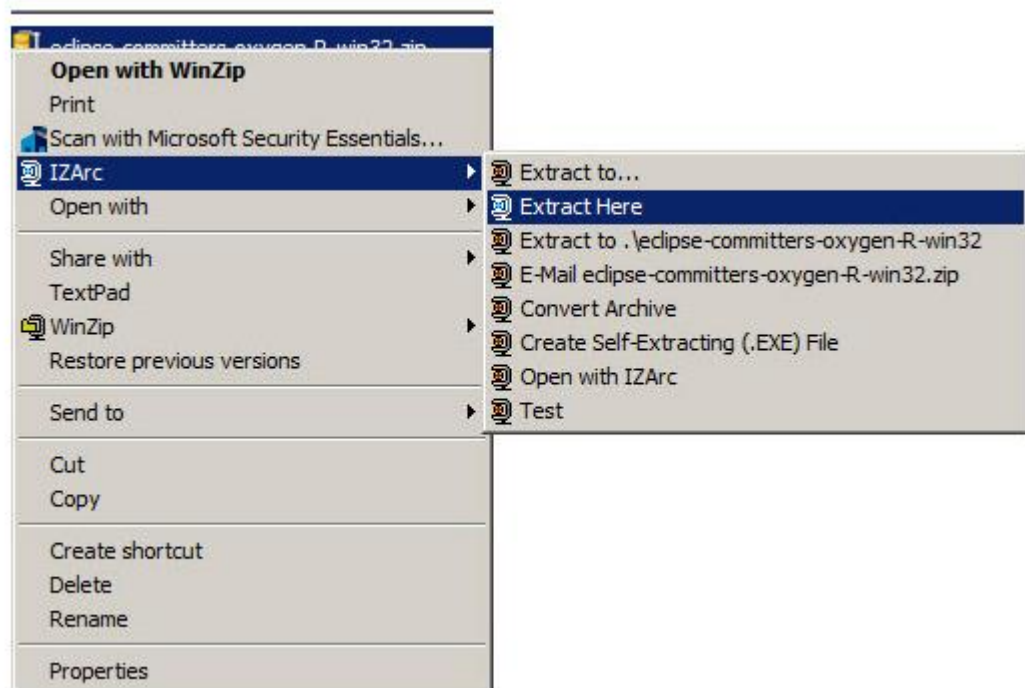
1. Move this file to a more permanent location, so that you can install Eclipse (and reinstall it later, if necessary).

2. Start the **Installing** instructions directly below.

Unzip **eclipse-committers-oxygen-R-win32.zip**, the file that you just downloaded and moved.

On my machine (running Windows 7), I can

- Right-click the file.
- Hover over the **IZArc** command from the menu of options.
- Click **Extract Here**



1. If you do not have IZArc or an equivalent unzipping program, here is the web site to download a free copy of [IZArc](#).

Unzipping this file creates a folder named **eclipse**; unzipping 250 MB can take a few minutes. You can leave this folder here or move it elsewhere on your hard disk. I recommend putting the downloaded file and resulting folder in the **C:\Program Files\** directory.

2. Create a shortcut on your desktop to the **eclipse.exe** file in this **eclipse** folder:

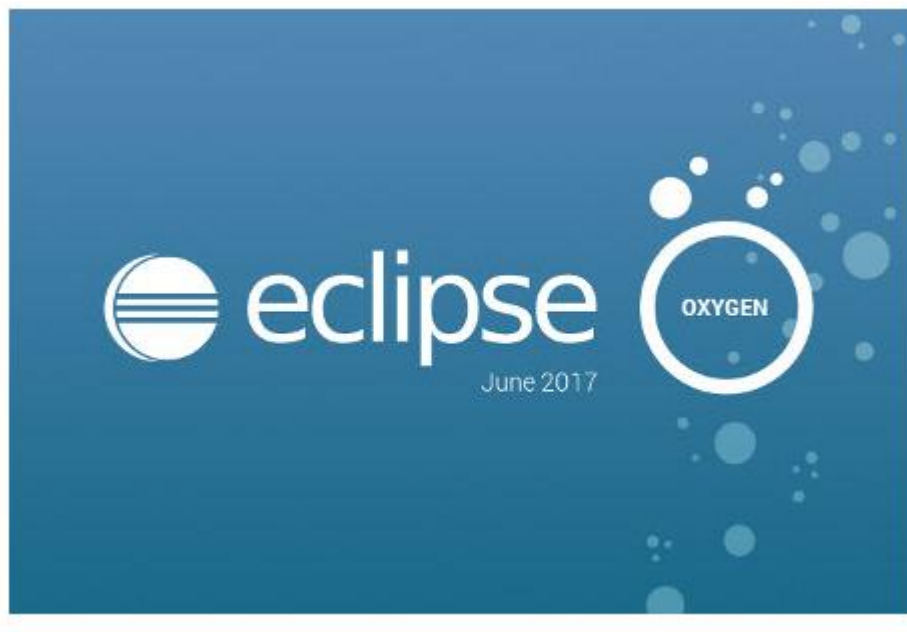
On most Windows machines, you can

- Right-click the file **eclipse.exe**
- Drag it to the desktop.
- Release the right button.
- Click **Create shortcut here**

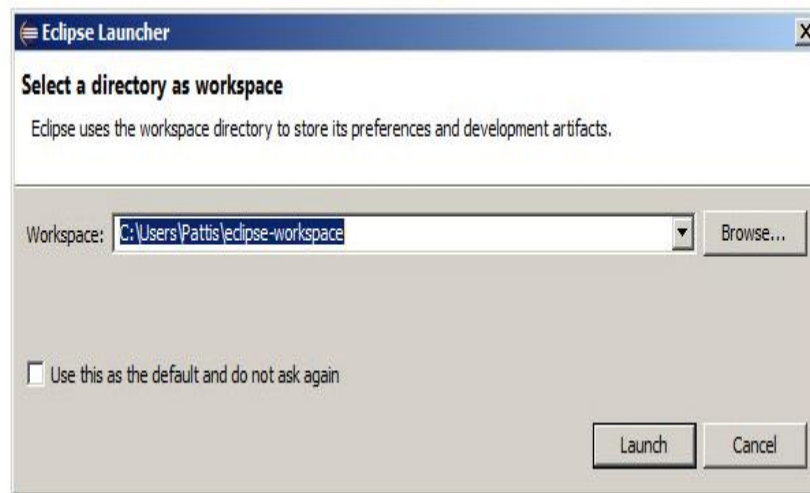
Now you are ready to perform a **one-time only** setup of Eclipse on your machine.

3. Double-click the shortcut to Eclipse that you just created above.

The following splash screen will appear



and then an **Eclipse Launcher** pop-up window will appear.



1. In the **Workspace** text box, your name should appear between **C:\Users\** and **\eclipse-workspace**, instead of **Pattis**.

Leave **unchecked** the **Use this as the default and do not ask again** box. Although you will use this same workspace for the entire quarter (checking projects in and out of it), it is best to see this **Workspace Launcher** pop-up window each time you start Eclipse, to remind you where your workspace is located.

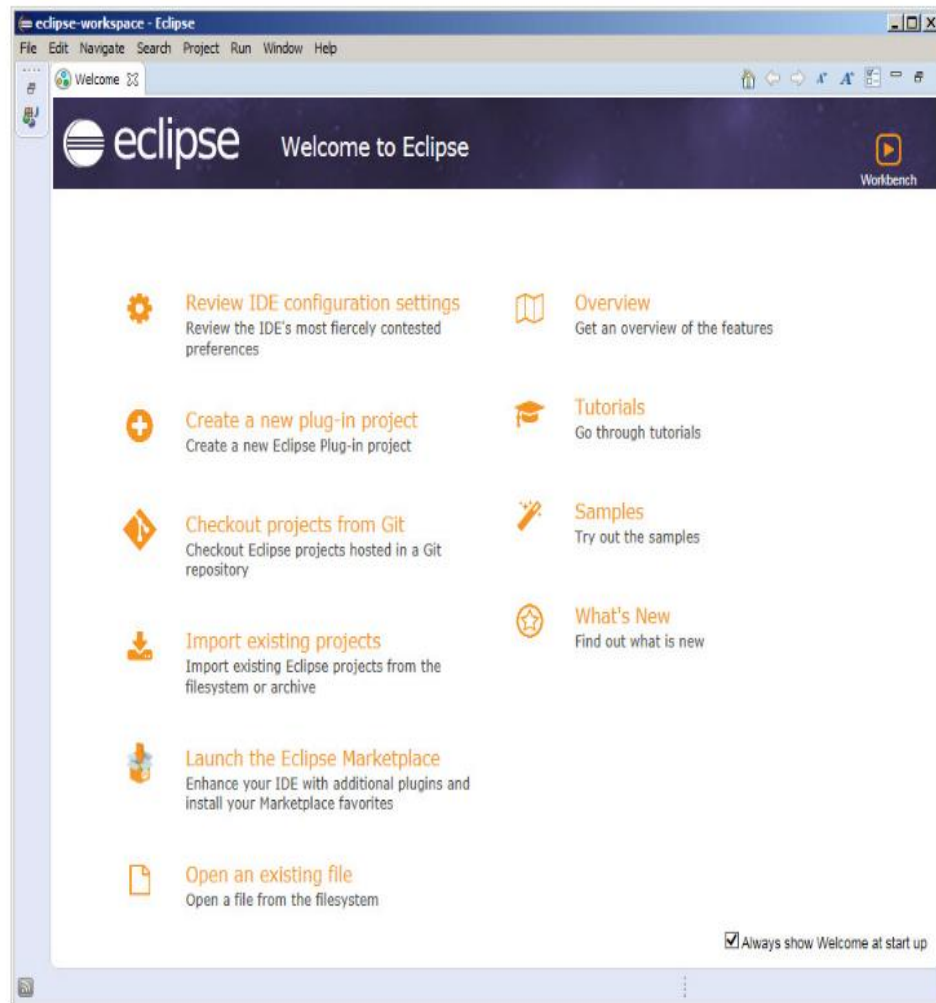
In fact, it is a good idea to create on your desktop a shortcut to your workspace folder; but you must click **OK** (see below) before Eclipse creates this folder and you can create a shortcut to it.

2. Click **Launch**.

Progress bars will appear as Eclipse loads.

Eventually the Eclipse workbench will appear with a **Welcome** tab covering it.



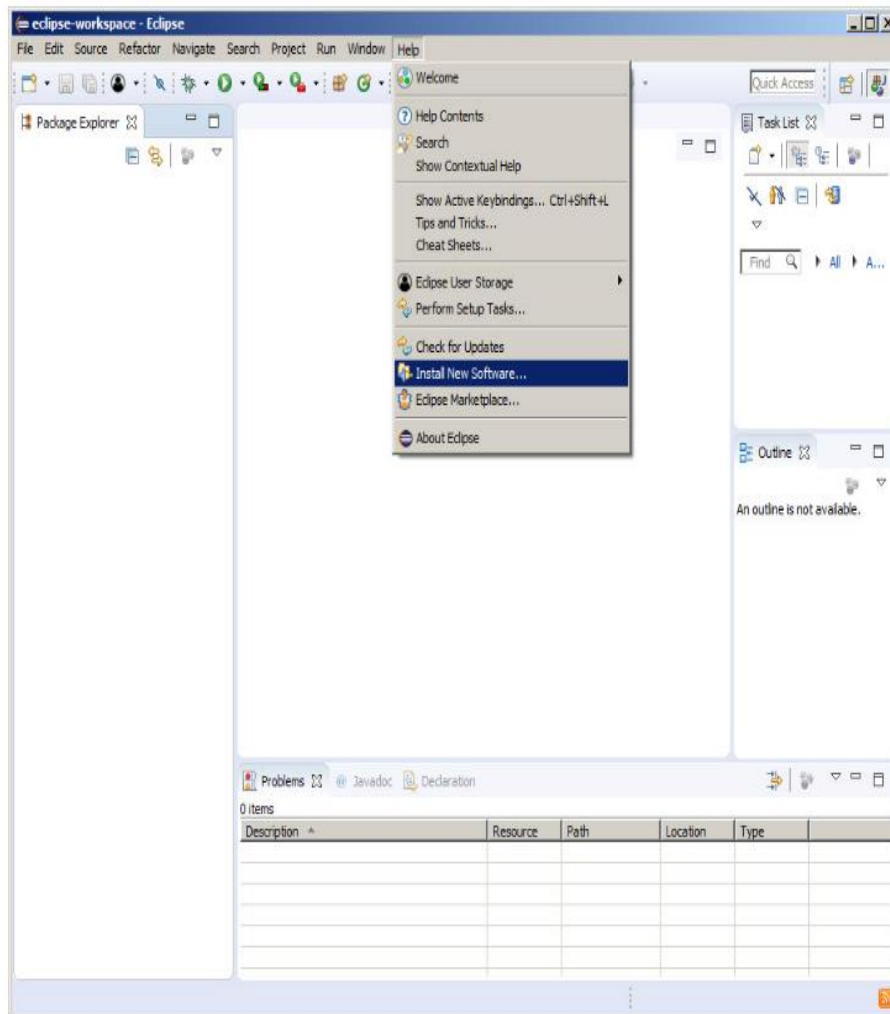


1. Terminate (click **X** on) the **Welcome** tab.

You will not see the **Welcome** tab when you start Eclipse again, after this first time.

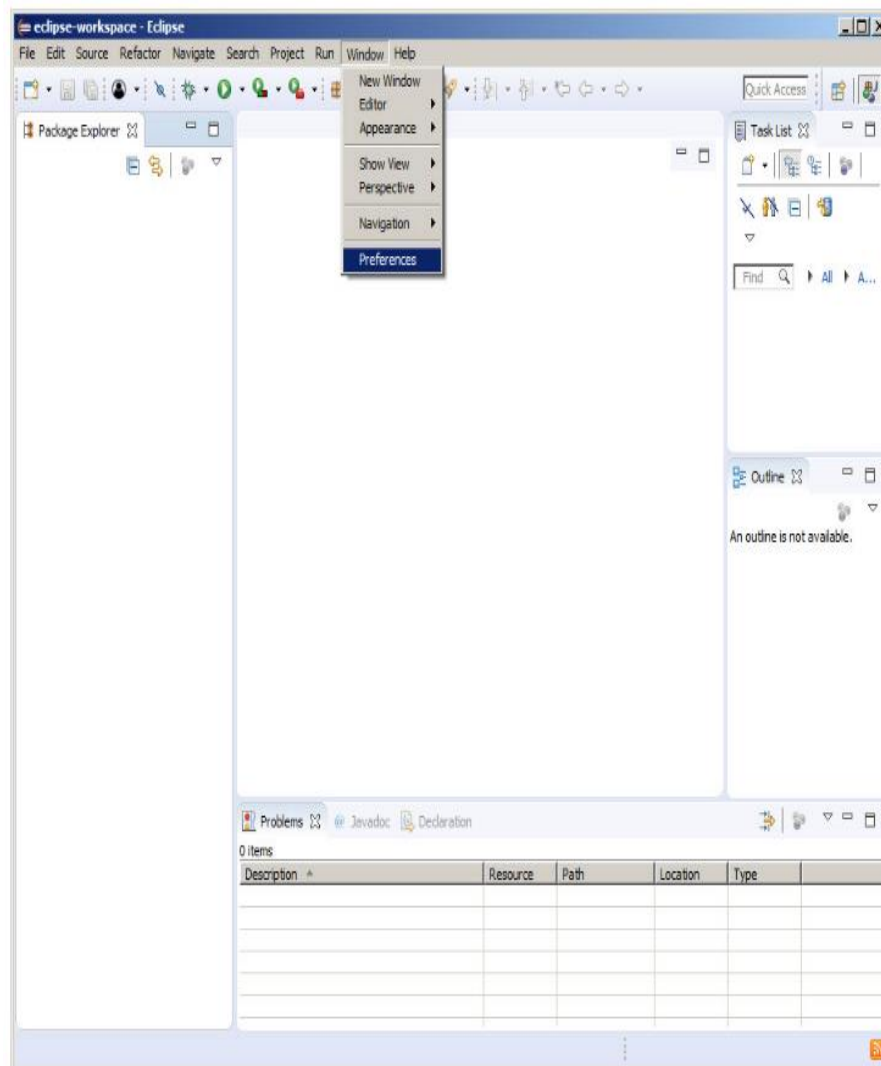
2. Click **Help** (on the far right of the line below this window's blue title **eclipse-workspace - Eclipse**) and then click **Install New Software...** in its pull-down menu, as shown below.





The Install pop-up window will appear.

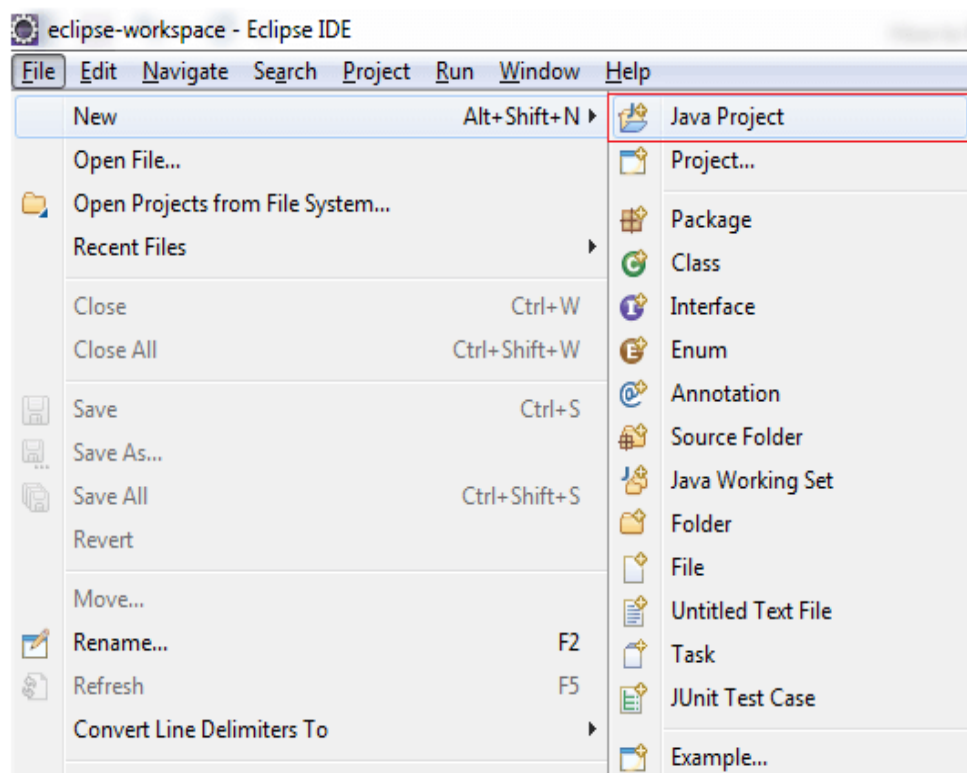
Click Window (to the left of Help on the far right of the line below this window's blue title eclipse-workspace - Eclipse) and then click Preferences in its pull-down menu, as shown below.

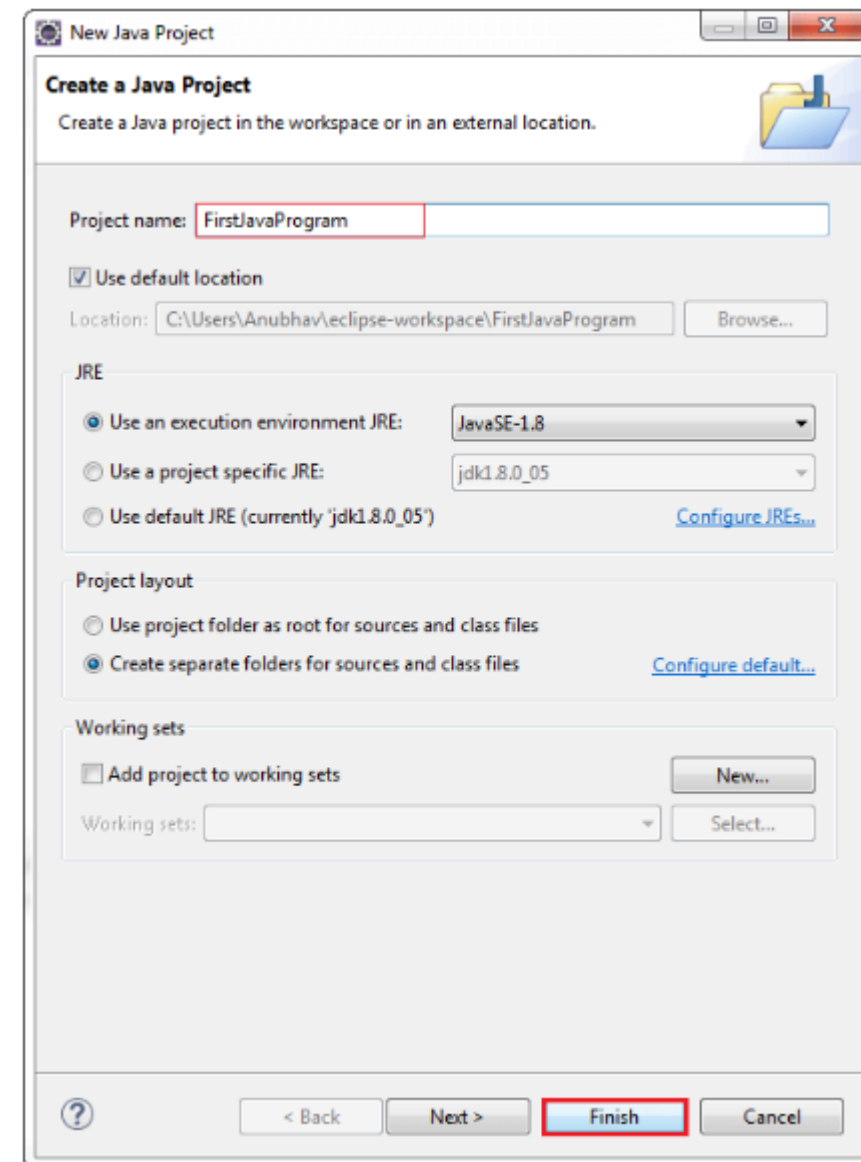


- Create a project.
- Right-click on your project and choose **Properties**.
- Click on **Java Build Path**.
- Click on the **Libraries** tab.
- Click on **Add External JARs...**
- Navigate to **junit.jar**. It should be in a location such as **...Eclipse 3.0.1\plugins\org.junit\_3.8.1\junit.jar**.
- Select **junit.jar**, click on **Open**, click on **OK**.
- To create a test class:

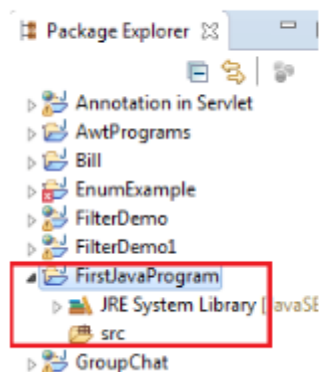
- Open the New wizard (**File > New > Other...**)
- Select **Java > JUnit** in the left pane and **TestCase** in the right pane and click **Next**.
- Enter the name of your test class and click **OK**.
- To run your test class, select your test class and choose **Run as > JUnit Test** from the **Run** drop-down menu in the toolbar.

**Step 1:** Open Eclipse and click **File > New > Java Project**.

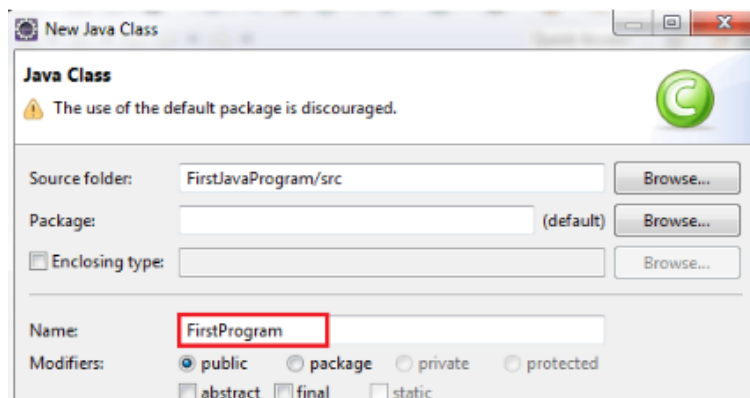




**Step 3:** In the Package Explorer (left-hand side of the window) select the project which you have created.



Step 4: Right-click on the src folder, select New > Class from the submenu. Provide the Class name and click on Finish button.



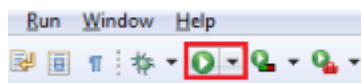
Step 5: Write the program and save it.

```

1
2 public class FirstProgram
3 {
4     public static void main(String[] args)
5     {
6         System.out.print("Hello! ");
7         System.out.print("Java");
8     }
9 }

```

Step 6: Now, press Ctrl+F11 or click on the Run menu and select Run or click on Run button.



Step 7: Output

```

Problems  Javadoc  Declaration  Search  Console  Servers
<terminated> FirstProgram [Java Application] C:\Program Files\Java\jdk1.8.0_05\bin\javaw.exe (Ju
Hello! Java

```

## STEPS FOR EXECUTING THE PROJECTS

### Step 1:

Open Eclipse and set the workspace

### Step2:

Right Click on the project Run As and Run On Tomcat Server

### Step3:

In middle we got tomcat error that time we need to change port number

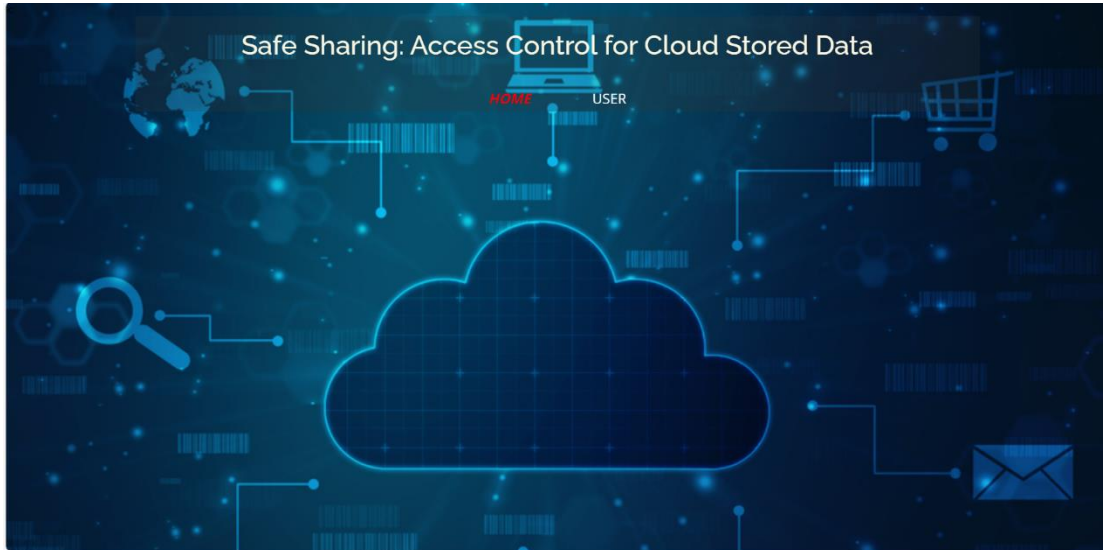
**Step4:**

Copy url in Google chrome and Run

## Chapter 7

### OUTPUT SCREEN SHOTS WITH DESCRIPTION.

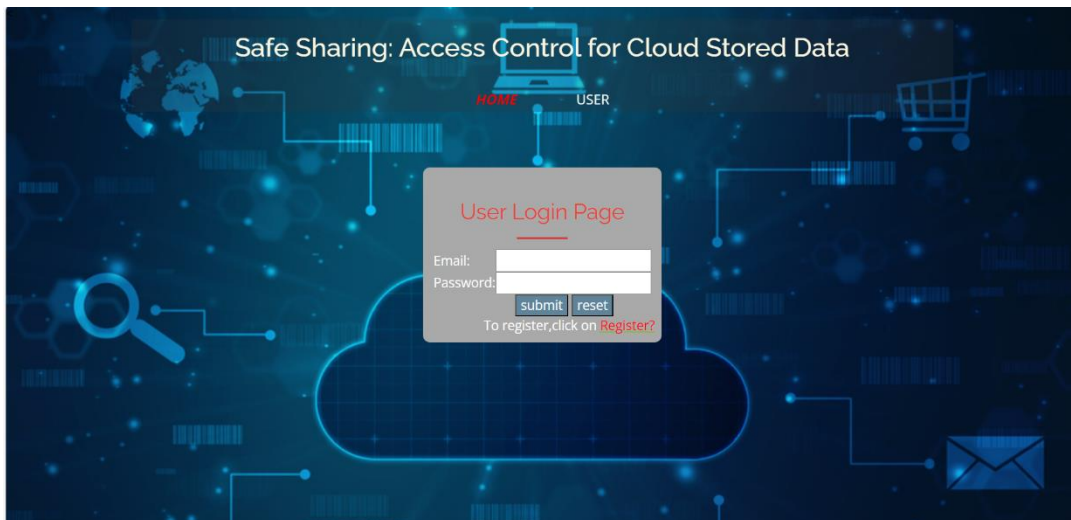
Home : this is the intial page of our project



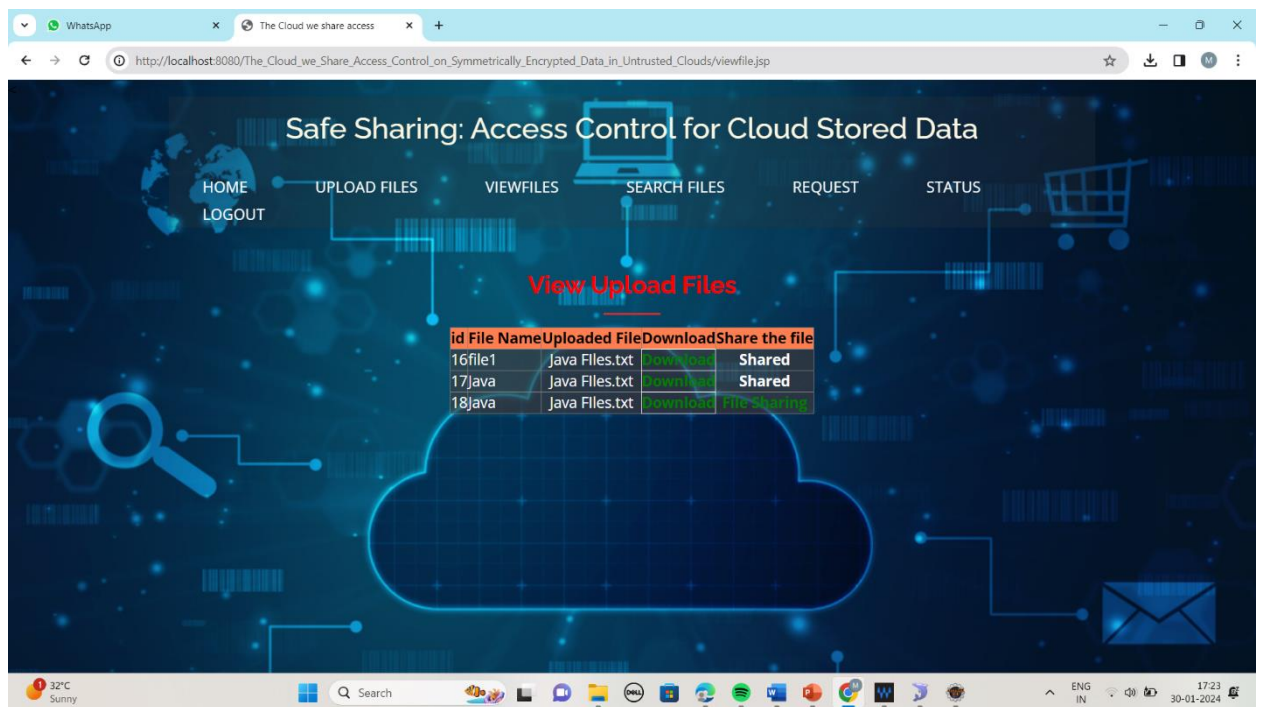
User Registration : Registration

The screenshot shows the User Registration form. The title 'Safe Sharing: Access Control for Cloud Stored Data' is at the top. Below the title, there are navigation links for 'HOME' and 'USER'. The background is a dark blue grid with various icons: a globe, a magnifying glass, a cloud, a shopping cart, and an envelope. The cloud icon is the largest and most prominent. The registration form is a light gray box with the title 'User Registration' in red. It contains the following fields: 'USER NAME', 'password', 'EMAIL ID', 'Mobile No', 'GENDER' (with a dropdown menu labeled 'select gendertype'), and 'ADDRESS'. At the bottom of the form are two buttons: 'Register' and 'Reset'.

User login : this User login page

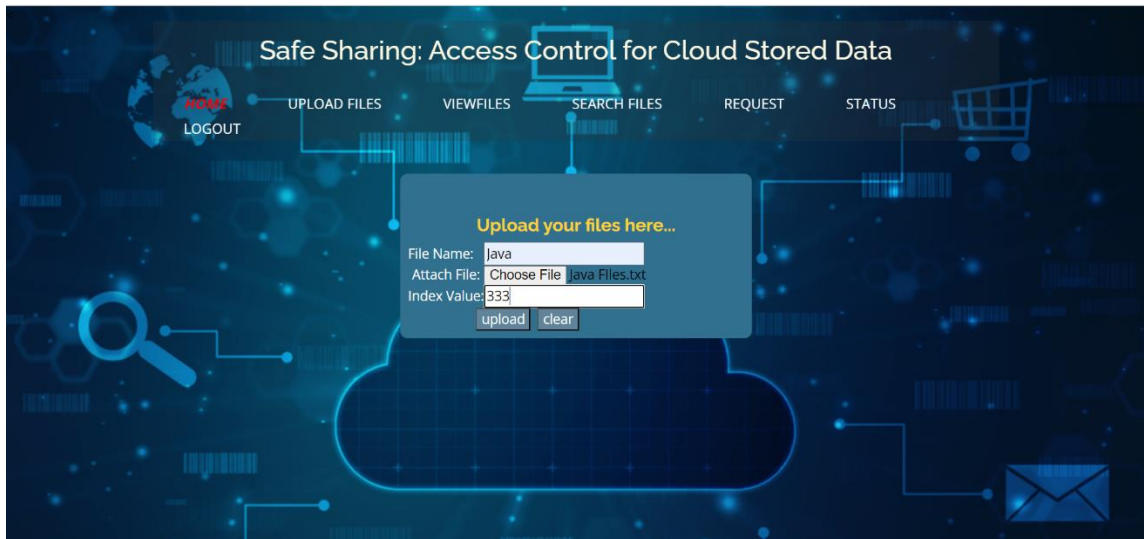


User home : User Home Page

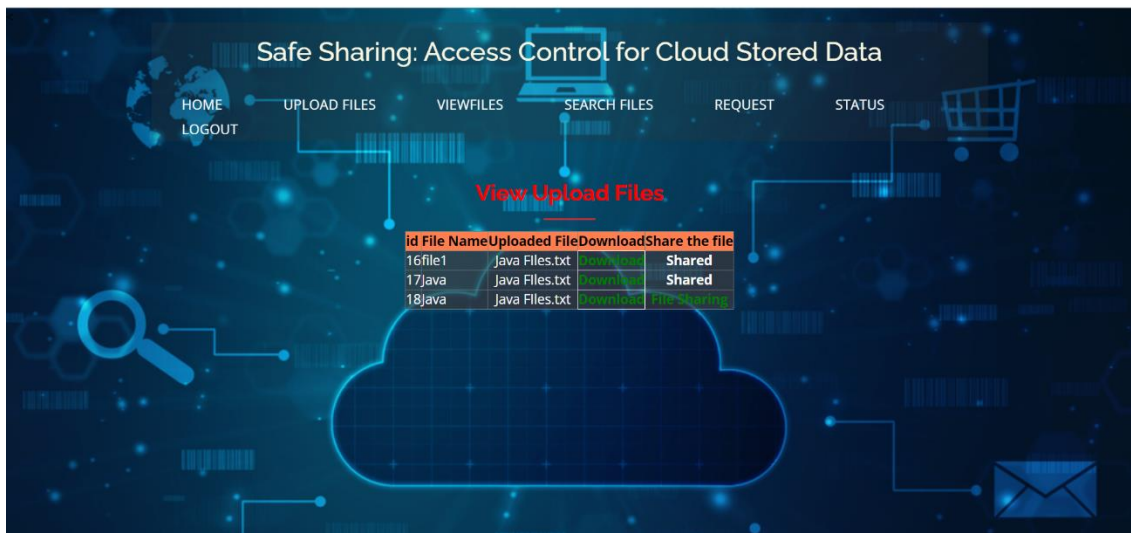


Upload : user Uploading the file.





User : User view the Uploaded files.

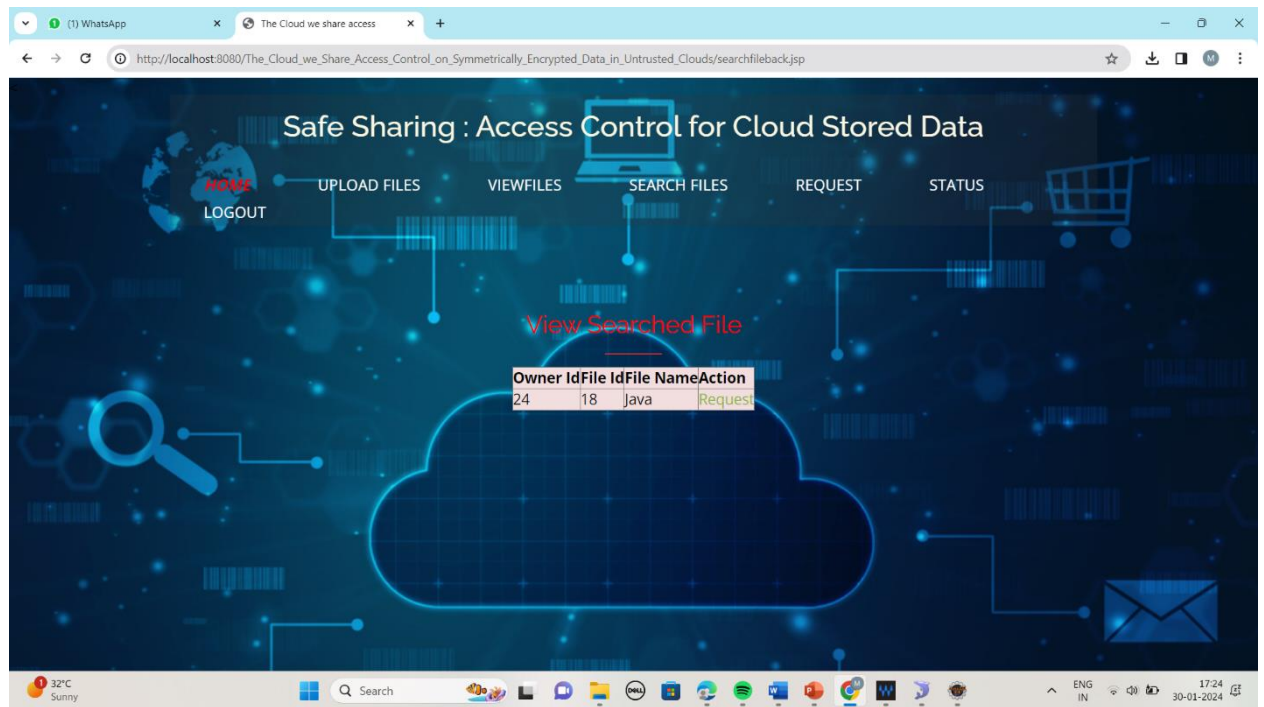


User : User view the searched file.

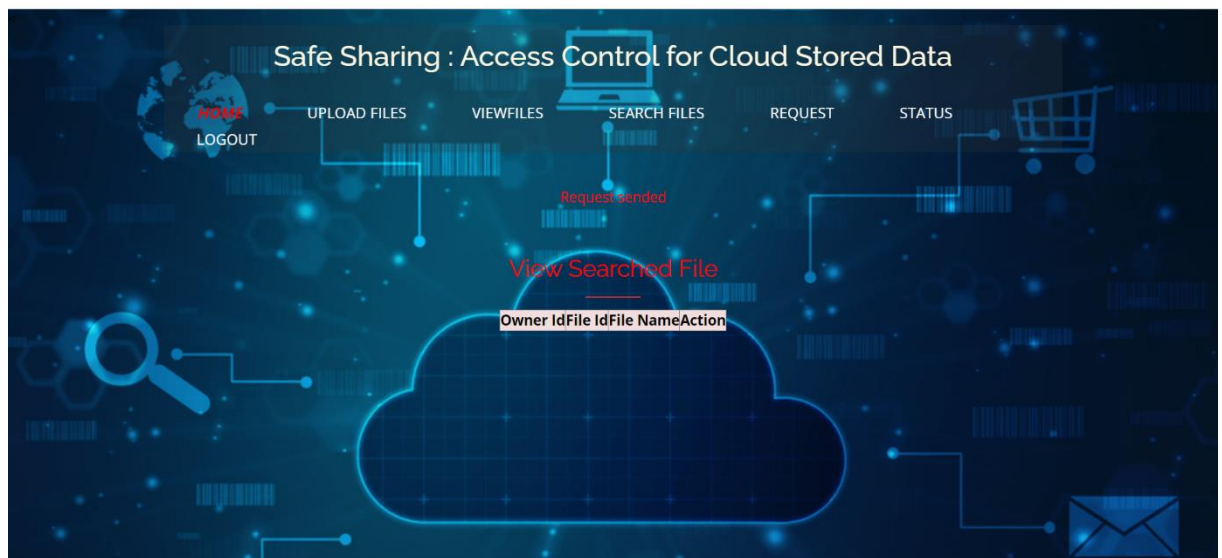
User : User view the searched file.

User : User view the searched file.

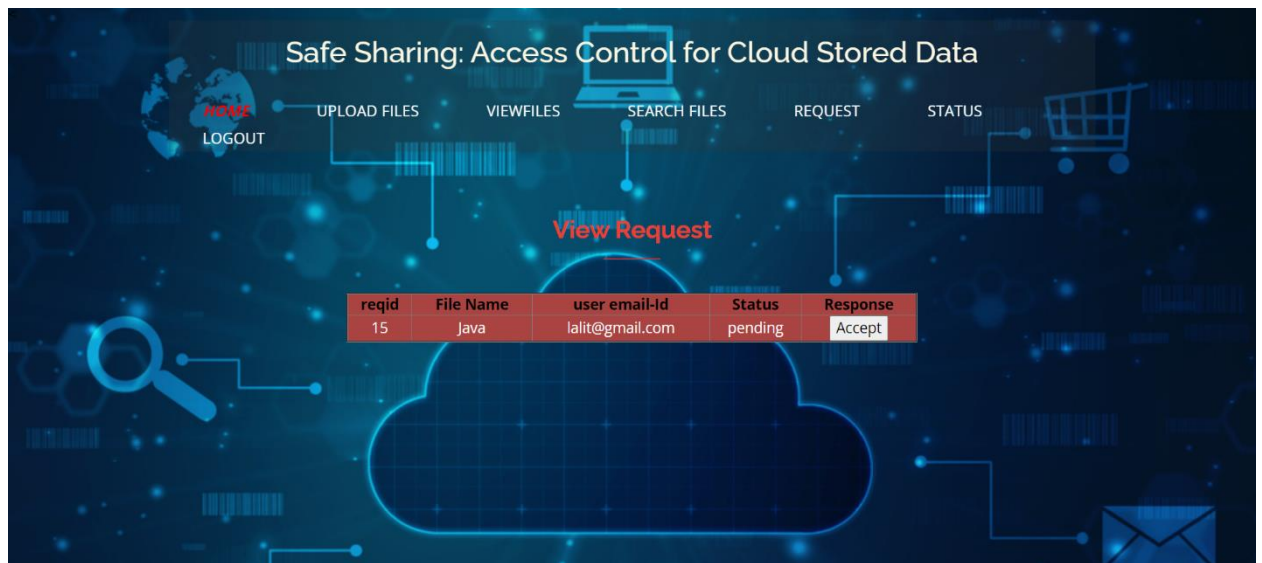
User : User view the searched file.



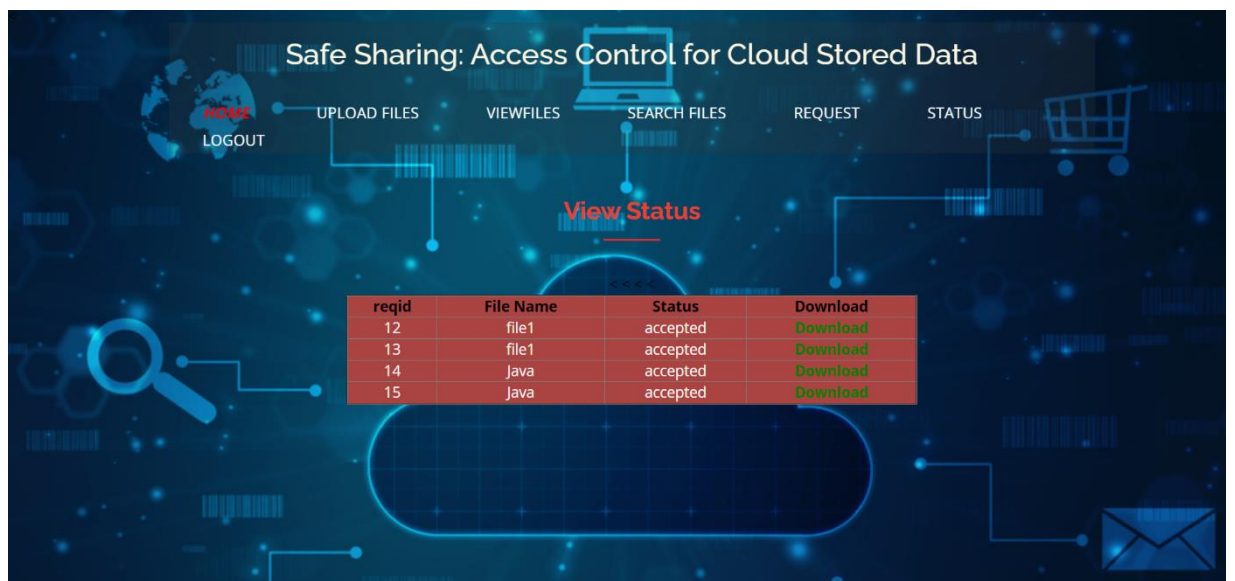
User : User sent the request for the file.



User : View the another user request.



User : user downloading the file.



## TEST CASES

S.NO	Test cases	I/O	Expected O/T	Actual O/T	P/F
1	User login	Valid/invalid credentials	Login success/ login failed	Login success/ login failed	P
2	User register	User details	If email not exist registration done else email already exists	If email not exist registration done else email already exists	P

**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

## CONCLUSION

In this paper, we proposed The Cloud we Share, a hybrid encryption scheme based on SSE and ABE. Our construction allows a data owner to share her data in a privacy-preserving way and manage the access rights of the rest of the users

**Future Scope:** In future we can implement to More security and provide Email Authentication.

## REFERENCES

- [1] S. Agrawal and M. Chase, “FAME: Fast attribute-based message encryption,” in Proc. ACM SIGSAC Conf. Comput. Commun. Secur., Oct. 2017, pp. 665–682.
- [2] G. Amjad, S. Kamara, and T. Moataz, “Forward and backward private searchable encryption with SGX,” in Proc. 12th Eur. Workshop Syst. Secur. (EuroSec). New York, NY, USA: Association for Computing Machinery, 2019.

- [3] A. Bakas and A. Michalas, “Modern family: A revocable hybrid encryption scheme based on attribute-based encryption, symmetric searchable encryption and SGX,” in *Security and Privacy in Communication Networks*, S. Chen, K.-K. R. Choo, X. Fu, W. Lou, and A. Mohaisen, Eds. Cham, Switzerland: Springer, 2019, pp. 472–486.
- [4] A. Bakas and A. Michalas, “Multi-client symmetric searchable encryption with forward privacy,” *Cryptol. ePrint Arch.*, Tampere Univ., Tampere, Finland, Tech. Rep. 2019/813, 2019. [Online]. Available: <https://eprint.iacr.org/2019/813>
- [5] A. Bakas and A. Michalas, “Power range: Forward private multi-client symmetric searchable encryption with range queries support,” in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, Jul. 2020, pp. 1–7.
- [6] J. Bethencourt, A. Sahai, and B. Waters, “Ciphertext-policy attribute-based encryption,” in *Proc. IEEE Symp. Secur. Privacy (SP)*. Washington, DC, USA: IEEE Computer Society, 2007, pp. 321–334.
- [7] A. Boldyreva, V. Goyal, and V. Kumar, “Identity-based encryption with efficient revocation,” in *Proc. 15th ACM Conf. Comput. Commun. Secur. (CCS)*, New York, NY, USA, 2008, pp. 417–426.
- [8] D. Boneh, X. Boyen, and E.-J. Goh, “Hierarchical identity based encryption with constant size ciphertext,” in *Advances in Cryptology—EUROCRYPT*, R. Cramer, Ed. Berlin, Germany: Springer, 2005, pp. 440–456.
- [9] D. Boneh, G. D. Crescenzo, R. Ostrovsky, and G. Persiano, “Public key encryption with keyword search,” in *Proc. Int. Conf. Appl. Cryptograph. Techn.* Berlin, Germany: Springer, 2004, pp. 506–522.
- [10] R. Bost, “ $\sigma$  οφος: Forward secure searchable encryption,” in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2016, pp. 1143–1154.
- [11] R. Bost, B. Minaud, and O. Ohrimenko, “Forward and backward private searchable encryption from constrained cryptographic primitives,” in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2017, pp. 1465–1482.
- [12] V. Boyko, “On the security properties of OAEP as an all-or-nothing transform,” in *Advances Cryptology—CRYPTO*, M. Wiener, Ed. Berlin, Germany: Springer, 1999, pp. 503–518,
- [13] V. Costan and S. Devadas, “Intel SGX explained,” *Cryptol. ePrint Arch.*, Intel, Mountain View, CA, USA, Tech. Rep. 2016/086, 2016.
- [14] R. Dowsley, A. Michalas, M. Nagel, and N. Paladi, “A survey on design and implementation of protected searchable data in the cloud,” *Comput. Sci. Rev.*, vol. 26,



pp. 17–30, Nov. 2017.

[15] M. Etemad, A. Küpçü, C. Papamanthou, and D. Evans, “Efficient dynamic searchable encryption with forward privacy,” *Proc. Privacy Enhancing Technol.*, vol. 2018, no. 1, pp. 5–20, Jan. 2018.

[16] B. Fisch, D. Vinayagamurthy, D. Boneh, and S. Gorbunov, “IRON: Functional encryption using Intel SGX,” in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur. (CCS)*, Oct. 2017, pp. 765–782.

[17] B. Fuhry, R. Bahmani, F. Brasser, F. Hahn, F. Kerschbaum, and A.-R. Sadeghi, “HardIDX: Practical and secure index with SGX,” in *Data and Applications Security and Privacy*, G. Livraga and S. Zhu, Eds. Cham, Switzerland: Springer, 2017, pp. 386–408,

[18] S. Lee, M. Shih, P. Gera, T. Kim, H. Kim, and M. Peinado, “Inferring fine-grained control flow inside SGX enclaves with branch shadowing,” in *Proc. 26th USENIX Secur. Symp.*, Victoria, BC, Canada, Aug. 2017, pp. 557–574.

[19] J. K. Liu, T. H. Yuen, P. Zhang, and K. Liang, “Time-based direct revocable ciphertext-policy attribute-based encryption with short revocation list,” in *Proc. Int. Conf. Appl. Cryptogr. Netw. Secur.* Cham, Switzerland: Springer, Jul. 2018, pp. 516–534.

[20] A. Michalas, “The lord of the shares: Combining attribute-based encryption and searchable encryption for flexible data sharing,” in *Proc. 34th ACM/SIGAPP Symp. Appl. Comput.*, New York, NY, USA, Apr. 2019, pp. 146–155.

[21] A. Michalas, “Text files from Gutenberg database,” Tampere Univ., Tampere, Finland, Tech. Rep., Aug. 2019. [Online]. Available: <https://zenodo.org/record/3360392#.X7fuas0zaUk>



