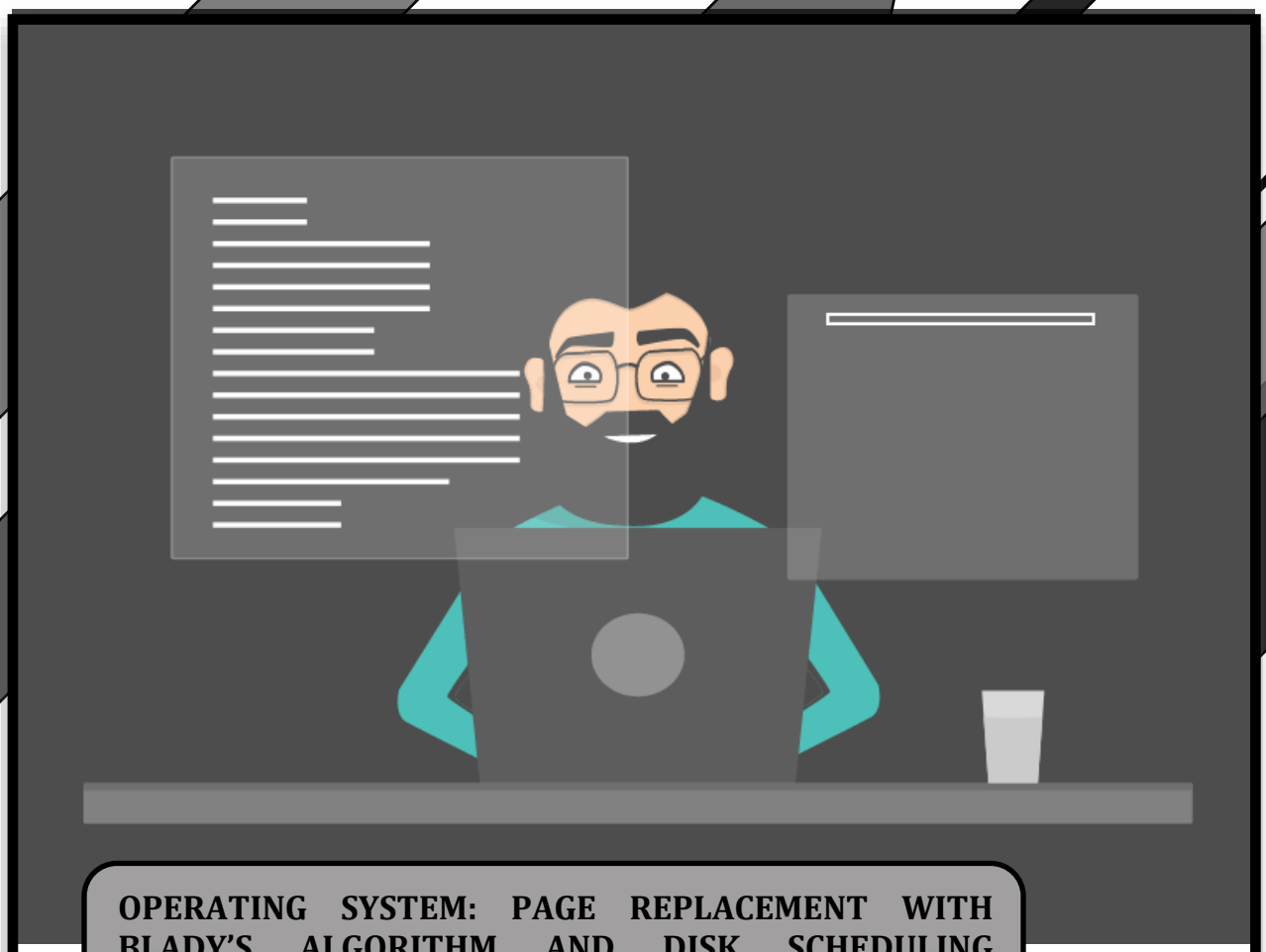


OPERATING SYSTEM: PAGE REPLACEMENT WITH BLADY'S ALGORITHM AND DISK SCHEDULING ALGORITHM

TEAM 14

- **BADAL PARMAR** **18BCP011**
- KHUSHI KATARIYA 19BCP068
- KALGI SHAH 19BCP062
- HETANSHI DABHI 19BCP053
- ADITYA GANDHI 19BCP002

TABLE OF CONTENT

⌑ ABOUT THE PROJECT:

1. Page Replacement Algorithm

Page Replacements algorithm is one of the most important concepts of the Operating System. The major drawback of any operating system is its speed and memory allocation process. The number of memory calls done to access frequently used pages, is tiresome and affects the speed, accuracy and durability of the computer hardware and the other software present in the system.

To overcome this issue, operating system came up with the concept of virtual memory management. Using the concept of storing the pages in the virtual memory, page replacement algorithms is introduced as a solution to paging issues faced in the operating systems.

2. Disk Scheduling Algorithm

Disk Scheduling is basically used to schedule I/O requests arriving in the system. It is important to schedule the I/O requests because at a time many I/O requests are made, and our disk controller can execute any one of the requests. So, to choose one request out of the others we use various disk scheduling algorithms so that each request is accessed in the minimal time and each request gets executed.

Before, moving on to the algorithms let's have a brief look at the core theory related to this concept:

A disk is a magnetic storage device which has several platters (or surfaces). The entire assembly of platters rotates as a unit at high speeds. The surface of each platter is organised as a concentric group of magnetic tracks on which data can be stored. Each track is divided into several blocks of fixed size in which the data is stored. A block is the smallest amount of data that can be read from or written to the disk in a single I/O operation.

Each disk surface has a read/write head which can move linearly across the surface. Data from the disk is requested by block number. The disk controller moves the head to the correct track and then waits for the correct block to pass underneath it for access.

In project the focus is on the **three** most important algorithms of the **Page Replacement** Topic and **6** algorithms of **Disk Scheduling Algorithms**. We are going to show how the algorithms are implemented, we are going to provide you with **algorithm calculator**, and **a comparison graph simulator**, that helps you to compare between all the algorithm for the similar kind of input.

⌑ IDE INSTALLATION:

For the successful implementation of the project, our team has used the **Eclipse IDE** and **Java Programming Language**. Eclipse IDE 2021-03 is a freely downloadable version of the world's most popular open source foundation that is supported by an active community of source developers and enthusiasts. Java Programming Language is the most efficient ad secure language and it has major framework or multiple packages for the GUI development.

To implement the codes and import the libraries we will need Eclipse IDE.

The installation process includes:

a) Install Eclipse IDE, version compatible above 4.13.

b) Install Java interpreter with eclipse, version compatible above jdk 11

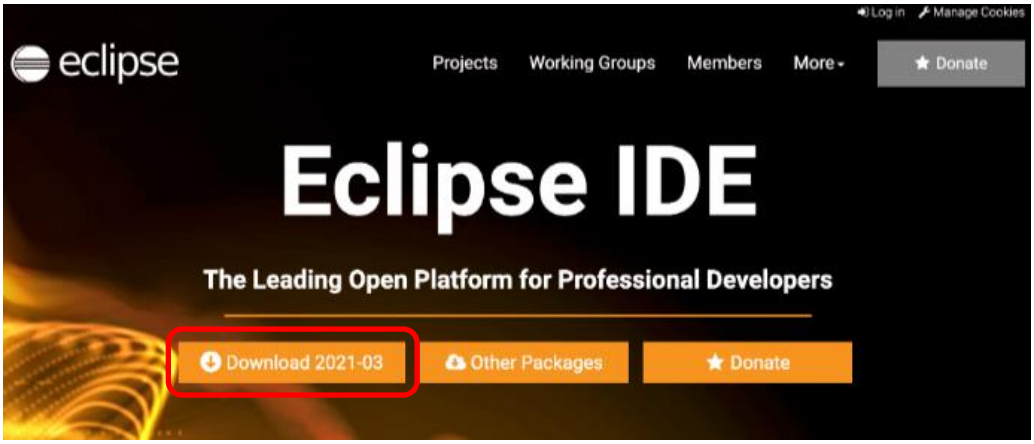
c) Build a java src workspace successfully.

d) Creating your first Java Project.

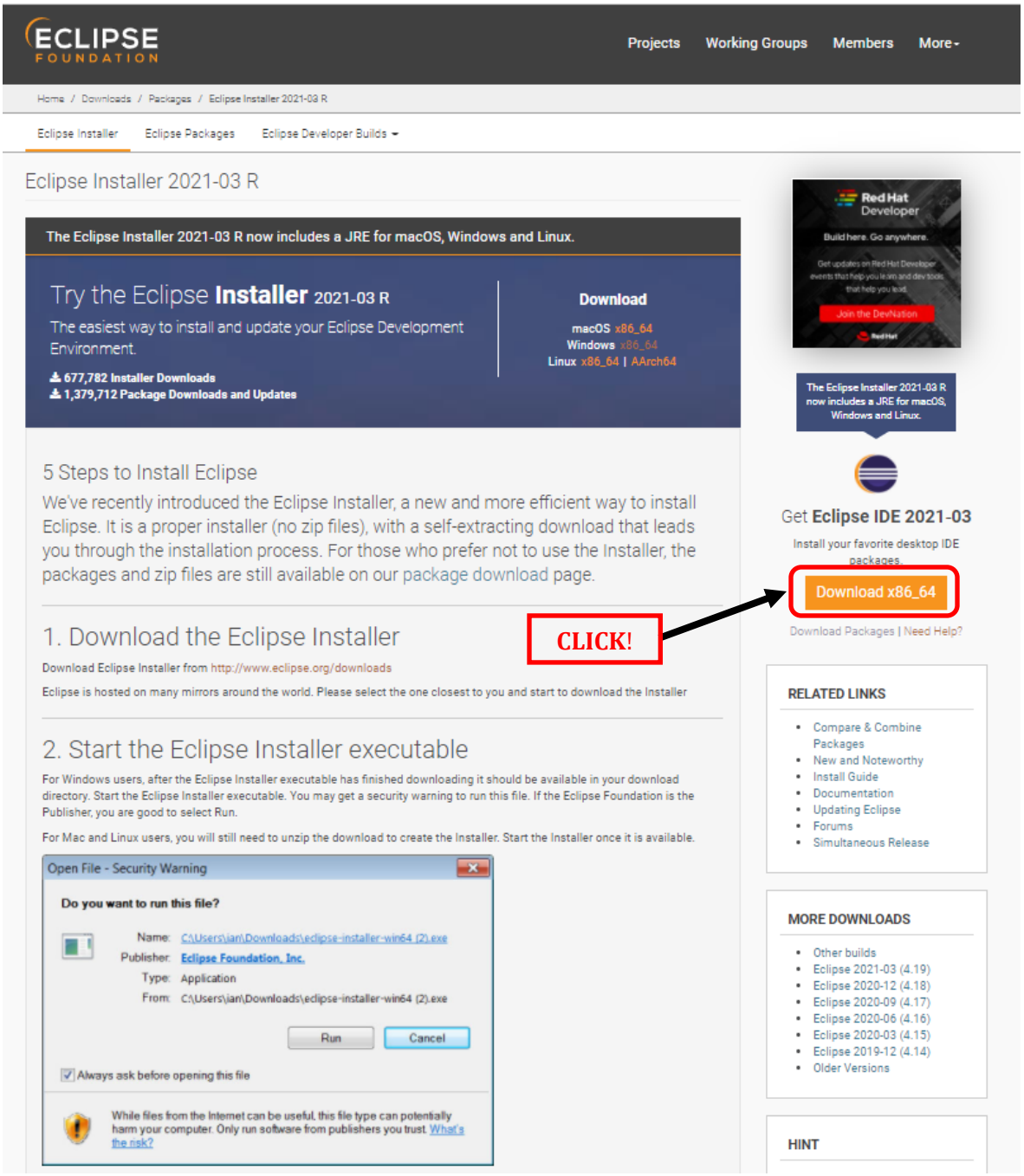
1. Installation of Eclipse IDE

   We have used Eclipse 2020-03(4.15) which is one of the latest versions of eclipse. There has advancement in this version, as every three months they update the system.

a. Go to the Eclipse 2020-03 homepage to download the software.
   OR
   Just go through the link below:
   Eclipse IDE 2020-03 | The Eclipse Foundation



b. After clicking the button, it will redirect you to another webpage, that describe all the steps for Installation
CLICK on the download button and you will be redirected to the download page then just follow the steps mentioned.

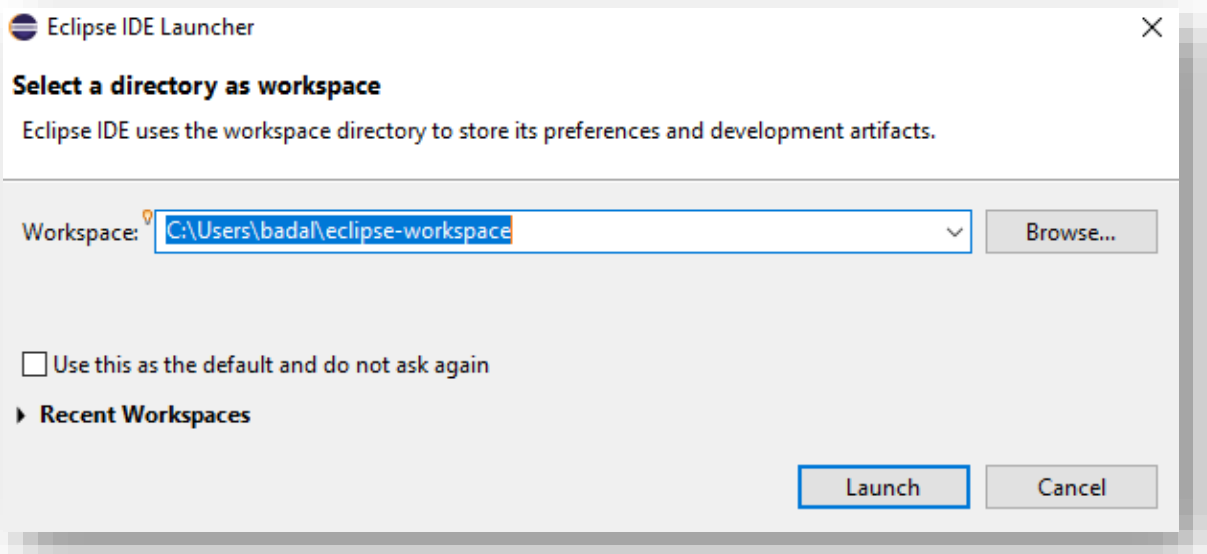


c. While installing the software, in the step 3 make sure you download Eclipse IDE for Java Developers or the 1st Option.

d. Save the eclipse software on your desired path/drive.

e. After installing the software, launch the software. Once you launch the software you will be asked to set up workspace.

LAUNCHING SOFTWARE



WORKSPACE SETUP

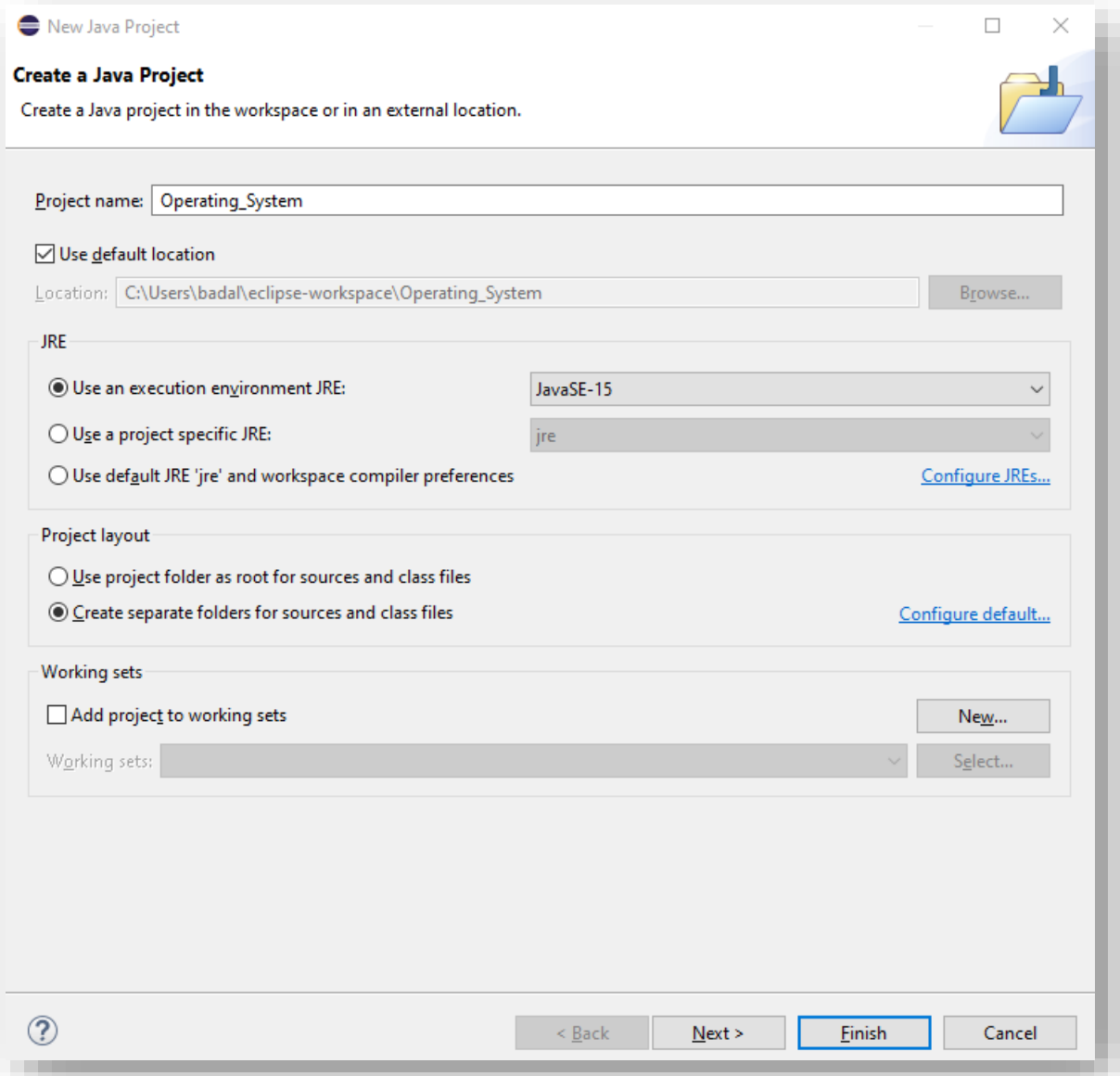
f. While setting up the workspace we would recommend you to setup the workspace in either E: drive or any other place apart from C: drive. This will help your heavy threads and error to be detected smoothly. Also, once the GUI is ready it will be easy you to run such heavy file seamlessly

g. Eclipse IDE is ready to run and now let create first running GUI project.

2. Creating First JAVA Project

   a. Launch the Eclipse IDE, you will receive a blank code area with a predefined default repository.

   b. Go to File -> New -> Project -> A dialog box will pop up



c. Name the Java Project as “Operating System” and save the file in your default repository/workspace and click “Finish”.

3. First JAVA project created now let us write some code into it. But to run this code we must import some external libraries that do not come predefined with the IDE. So, let’s download those libraries and then set a class path for it.

4. Still not able to understand the downloading or installation of Eclipse IDE just click on the link below:
[How to install Eclipse IDE 2021-03 on Windows 10 for Java Development - YouTube]

⌑ LIBRARY INSTALLATION

For this project we have used 3 external libraries, out of which 1 can be directly downloaded from the Eclipse IDE interface and other two can be downloaded from their respective web pages and then attach their class-path to our Operating System

Workspace.

1) Installing SWT Designers kit:

This kit is eminent part of our project. We need to download this kit to use the Windows Builder feature provided by the Java Developers and can create UI of our desktop application with Drag and Drop method and then the parser parses the generated codes for us. These methods save a lot of time rather than writing a tedious code.

a. **To check whether you have pre-installed SWT Designers kit.**
Go to **File** -> **New** -> **Other**. The "select a wizard" window pops up.

- Double click in Window Builder folder and then to Swing Designer subfolder.
- Click to the Application Window and then click Next.
- Give a Name for your new window and then click Finish. The Source code of the new Window pops up.
- Press Run (the "Play" icon of the toolbar) to run your newly created window.

b. **If you don't have pre-installed SWT Designer kit**.
Go to **Help** -> **Install New Software** -> From the **work** with drop, select any one Option -> Find for **SWT Designer** and install all the files.

SWT designer is successfully installed.

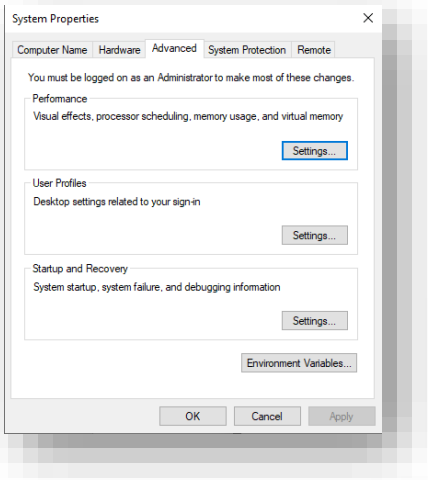
2) Installation for JfreeChart

The JFreeChart kit or set of jar files is one of eminent part of the visualization portion of our project. This is an opensource .jar file created by the Java Developers for the ease of the graph visualizations and dataset analysis and comparison charts etc.

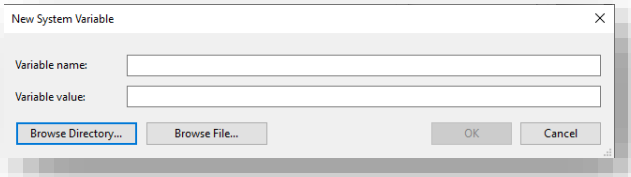
The method to download the JFreeChart .jar file. Go to **www.jfree.org** -> Click on the download tab It will redirect you to the GitHub repository -> Click on the file with (.zip) extension -> Save the downloaded file.

Extract those (.zip) file -> Extract it at the place where you have stored your project for easy fetching.

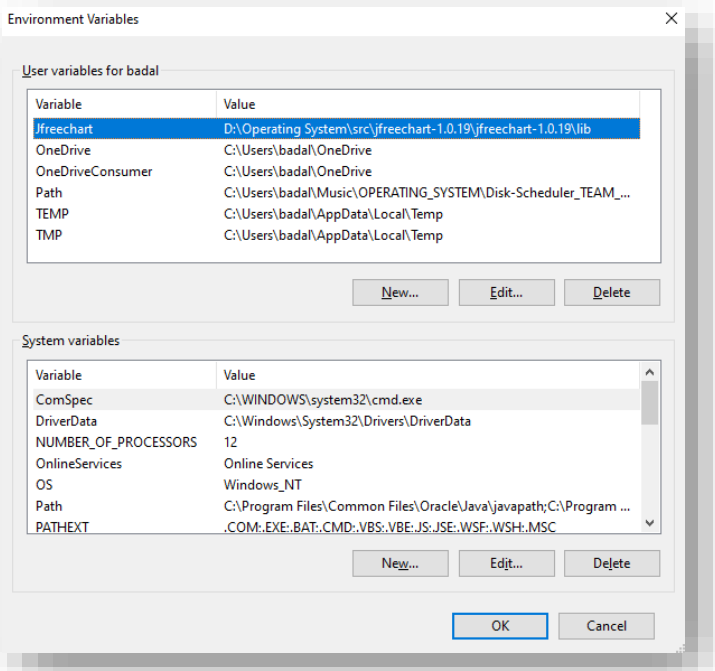
3) Building the path for this reference libraries



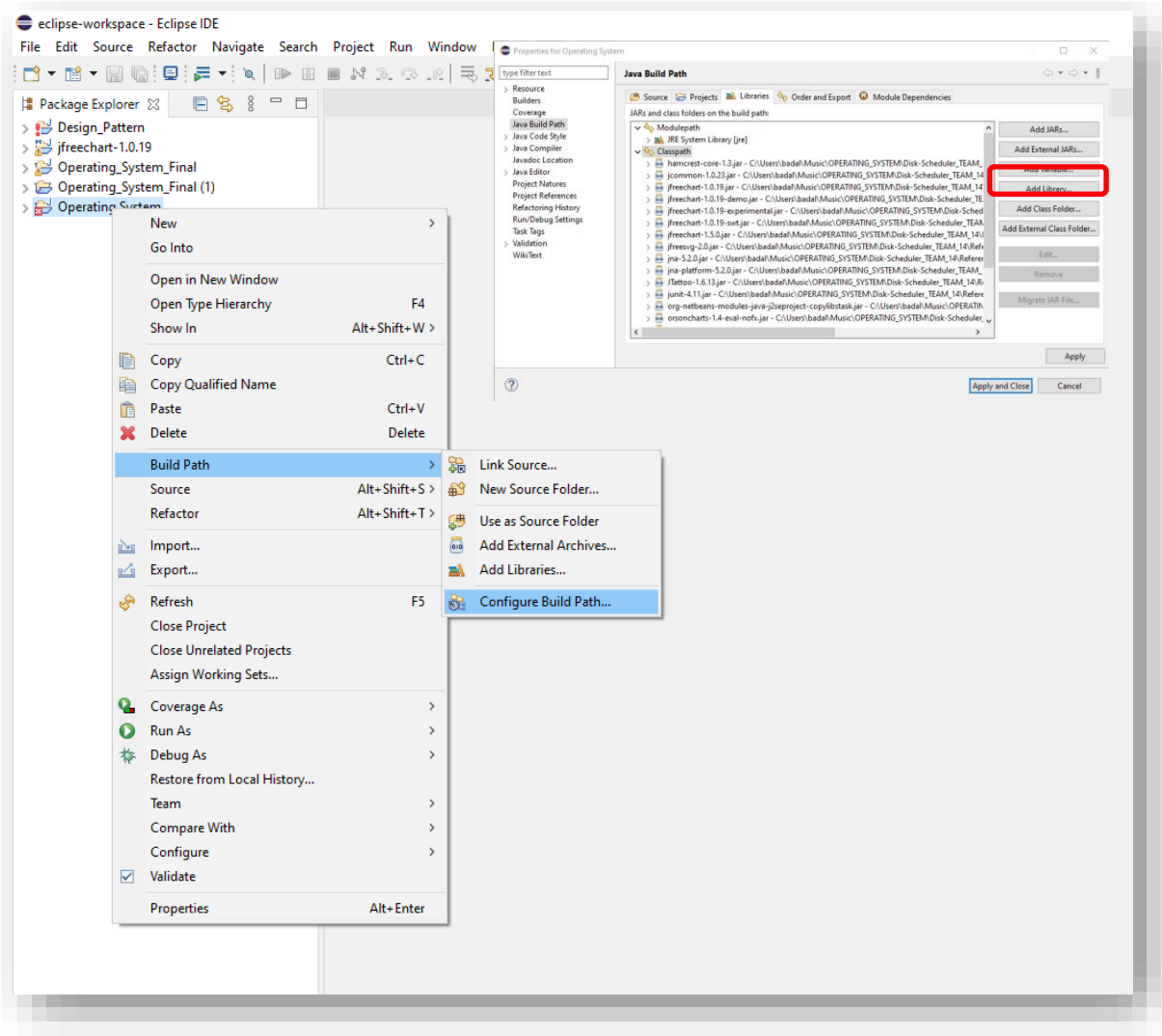
a. Search for Environmental Variables in the search tab of your PC
A Screen like below will show up, click on the Environment Variables



b. Choose Path -> Click new -> Go to place where you have extracted your files -> select jfreechart-1.0.19 -> select jfreechart-1.0.19 -> select lib -> click on Open.



c. Now launch Eclipse IDE -> Right Click the project -> Click Build Path -> Click Configure Build Path



d. A dialogue box appears -> Click on Add External Jar Files -> Execute the same process as we did for setting up the Environment Variable Path.

e. Voila, your JFreeChart is added to the project to be use. For crosscheck, Click the dropdown arrow on the left of the project -> Click the drop-down arrow on the left of Referenced Libraries -> JFreeChart Libraries would have been installed.

4) Installing and Building path for vlcj libraries

Download vlcj.jar file (Version 4.7.0) and Launch Eclipse IDE -> Follow the same steps that you did to build path for the JFreeChart -> Just make sure that this time you are attaching the vlcj files and not the JFreeChart files. All the libraries are set up and your project is ready to extract all the .java files now.

Other than, above mentioned External .jar files we have used some inbuilt libraries:

- **Javax Swing Package** for creating and viewing desktop app
- **Java AWT Package** for drawing and setting dimension and external layout in the Desktop App
- **Javax Windows Builder** method to create the visuals in the GUI
- **Java language Exception** package to handle all the possible exceptions in the code.
- **Java language Thread** class to change between multiple JFrames smoothly and effortlessly.

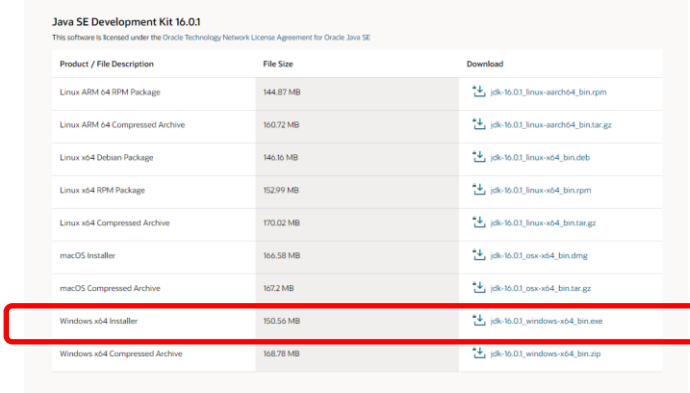
Extract and Run:

I. Now you have imported and attached all the libraries that will be needed to run the project.

II. Now extract the .zip file -> Go to src folder -> Copy paste all the files to the src folder of your project.

III. Launch the **Eclipse IDE** -> Compile all the codes one by one -> Check whether there is some error or not -> **Run**

IV. An compiled application format (.exe) and a Runnable Jar file is provided for compilation without running any codes (for easy accessibility)
To run the .exe or .jar file one needs to install **JRE (Java Runtime Environment),** if not installed, install it from link below:
Java SE Development Kit 16 - Downloads (Window x64 installer– 150.56 Mb file)



⌑ CONCLUSION

All the above-mentioned codes are running properly without any errors.

THINGS LEARNED THROUGH THIS PROJECT:

Through teamwork we can achieve our goal in limited time.
How to implement different libraries of java and connect all the frames to each other for seamless graphical user experience.
The entire concept of Disk Scheduling Algorithm thoroughly.
Implement animations and Desktop App with video, voiceovers, charts, graphs, and content.

We want to collectively thank Chintan sir and Samir Sir for giving us this eminent opportunity to create this GUI and guiding us throughout out project. We are pleased to announce that we have completed our work and will be waiting for your suggestions and constant support in the coming future.

⌑ REFERENCES

https://www.geeksforgeeks.org/ (For descriptive information about Page Replacement and Disk Scheduling Algorithms)

https://www.flaticon.com/authors/flat-icons (For all the transparent icons used n GUI)

https://www.jfree.org/jfreechart/ (For graph in GUI)

https://capricasoftware.co.uk/projects/vlcj (For video and voice over animations GUI)

https://www.eclipse.org/eclipseide/ (For Eclipse IDE 2020)

http://www.jtattoo.net/Download.html (For GUI Theme)

Note: **.exe** file and **.jar** file are to be run/compiled just by installing JRE package for accessing the GUI program.