Experiment No - 08

Author Name	Karan Agrawal
Roll No.	39
Sem & Sec	7 CSE [B]
Date Compiled	12-Oct-2023
File Name	Prac8

AIM: Demonstrate your understanding of cloud computing concepts and your ability to use CloudSim for simulating a cloud environment to simulate the provisioning of virtual machines, resource allocation, and workload execution. (CO1, CO4)

Perform the following tasks on CloudSim:

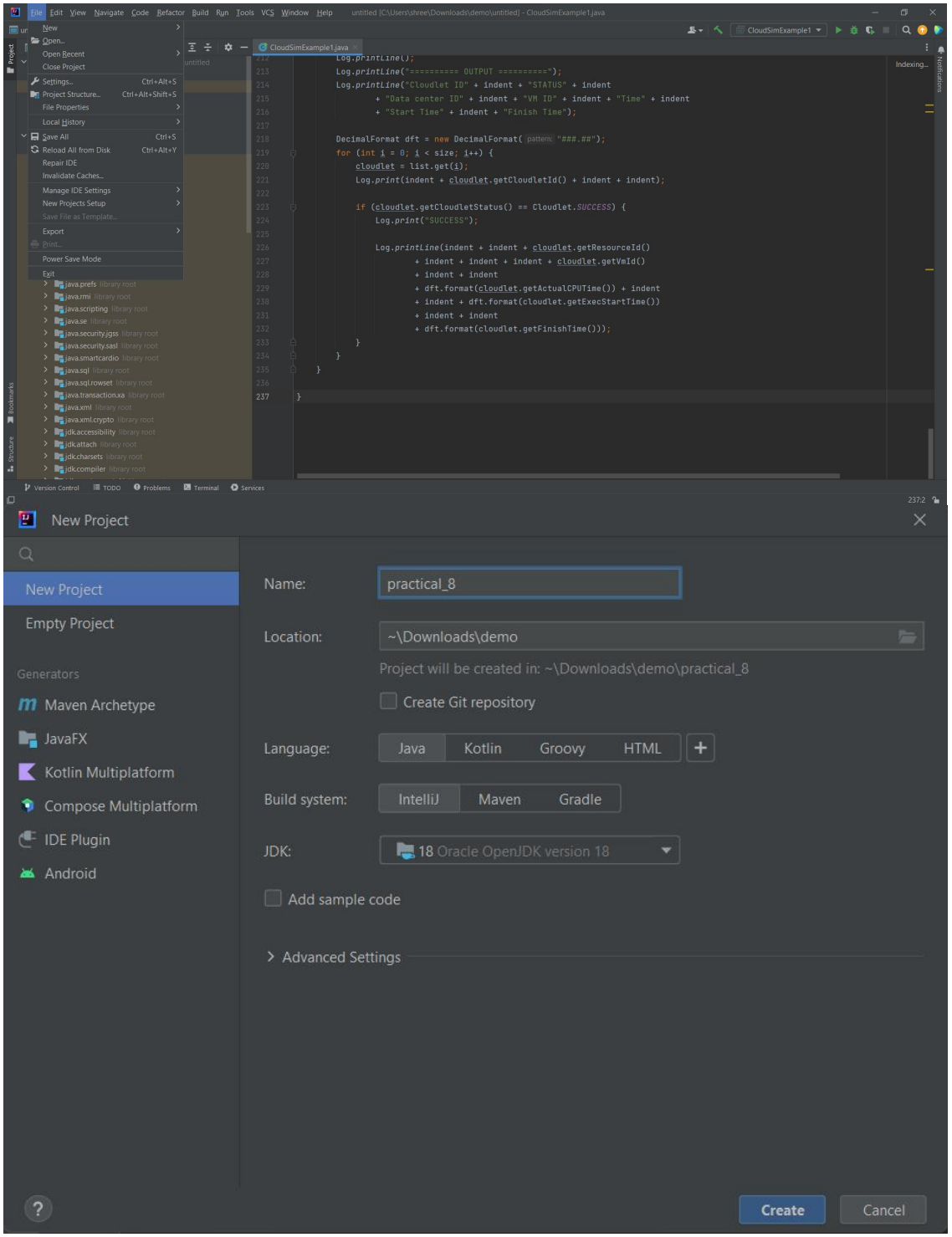
1. Setup of the CloudSim environment.

2. Monitor creation and provisioning of virtual machines, resource utilization and result simulation and analysis for various paramenters using given examples.

3. Also, understand and implement load balancing algorithms via Cloud Analyst in CloudSim environment.

Step 1: From the zip folder extracts cloudsim-3.0.3 into a folder. Also, extract the commons-math3-3.6.1 jar into the same folder.

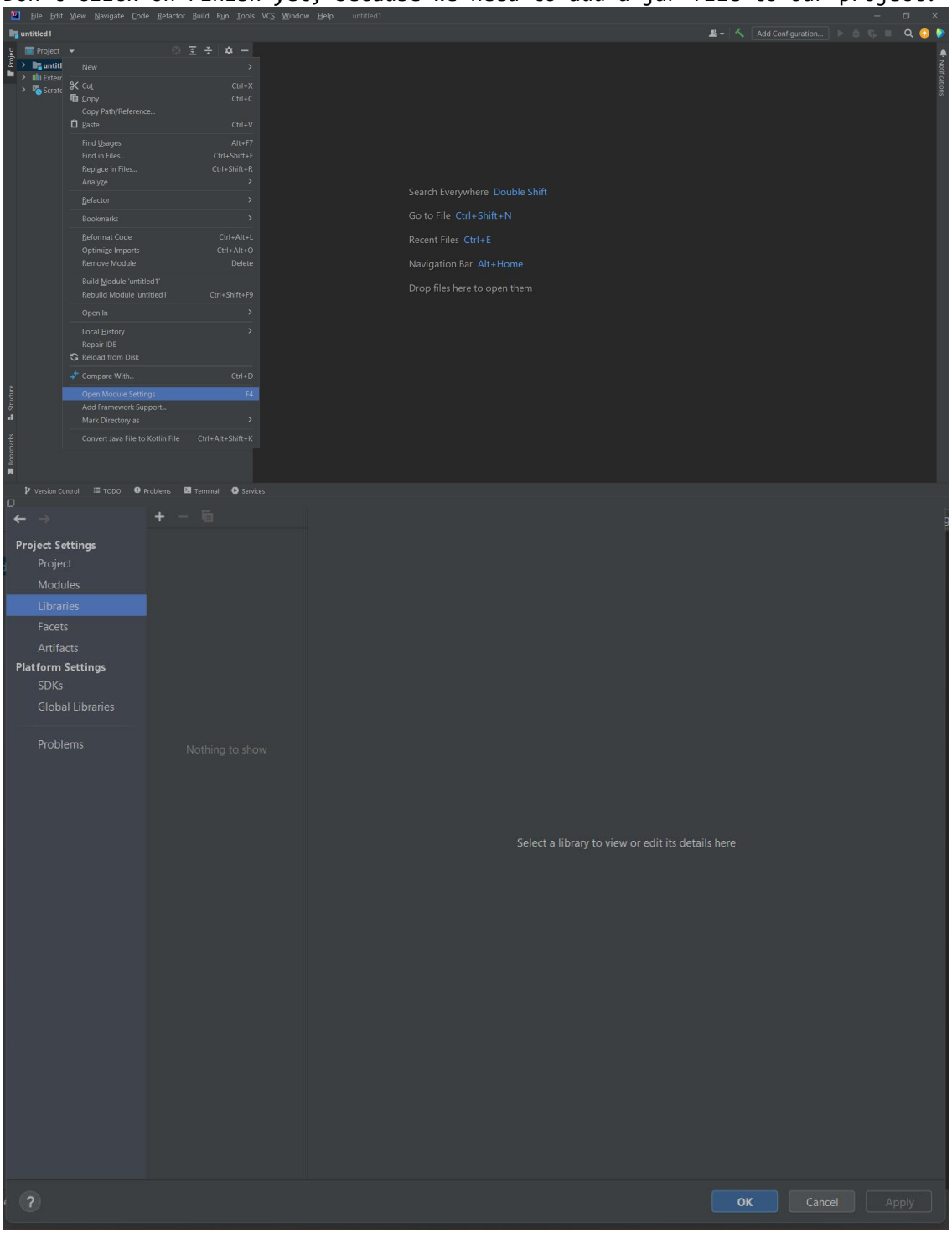
Step 2: Open Eclipse IDE and go to File -> New -> Java Project.



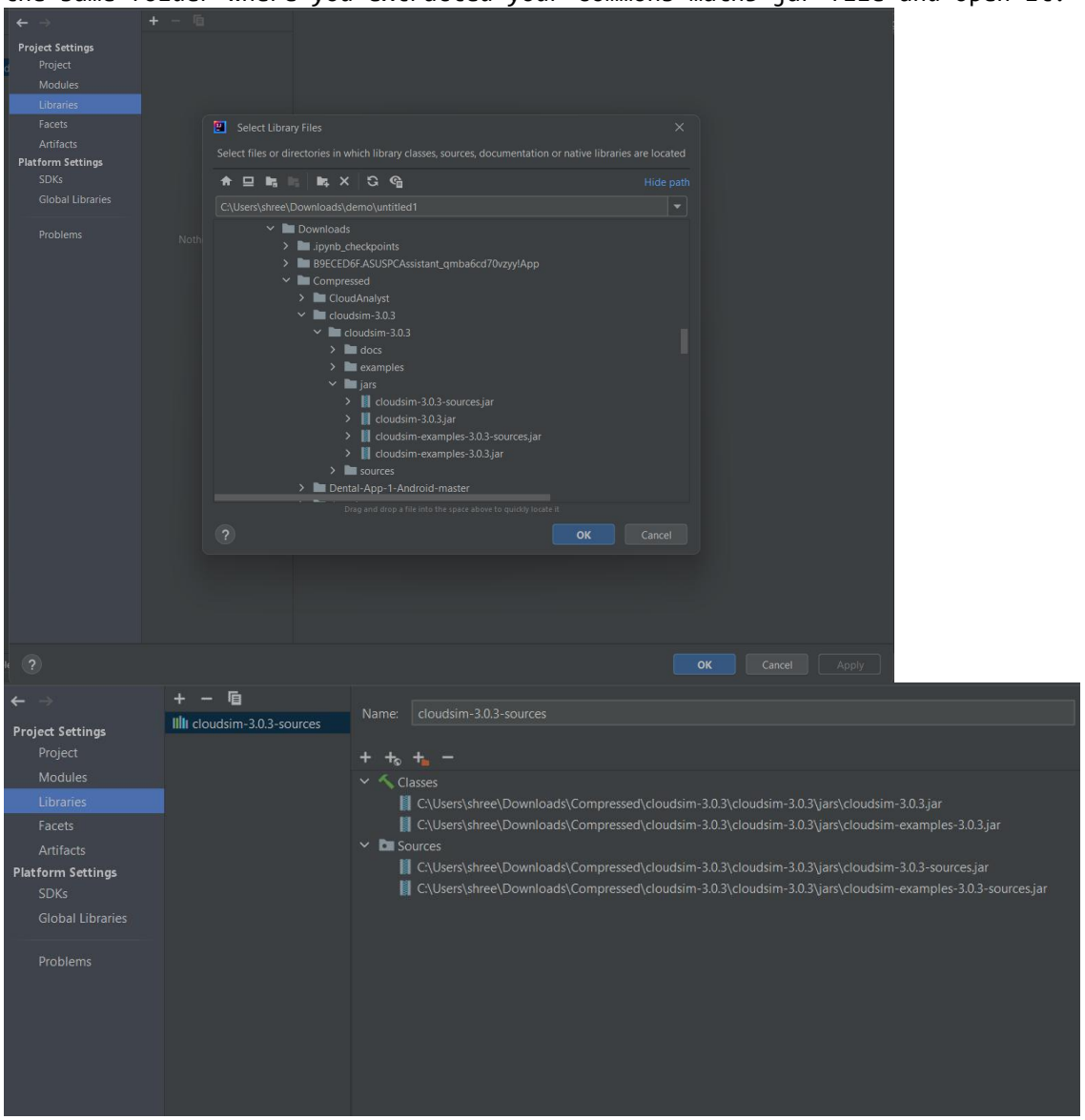
Step 3: Enter any name for your project and then uncheck the Use default location box just under it and click on Browse.

Browse to the folder where you extracted your files and select the cloudsim-3.0.3 folder.

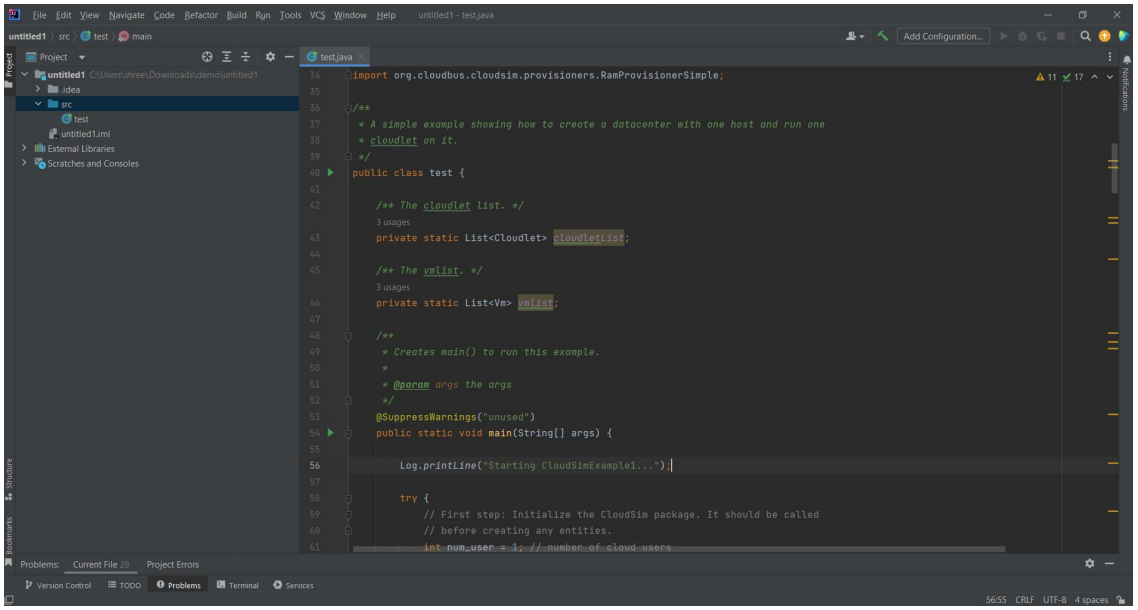
Don’t click on Finish yet, because we need to add a jar file to our project.

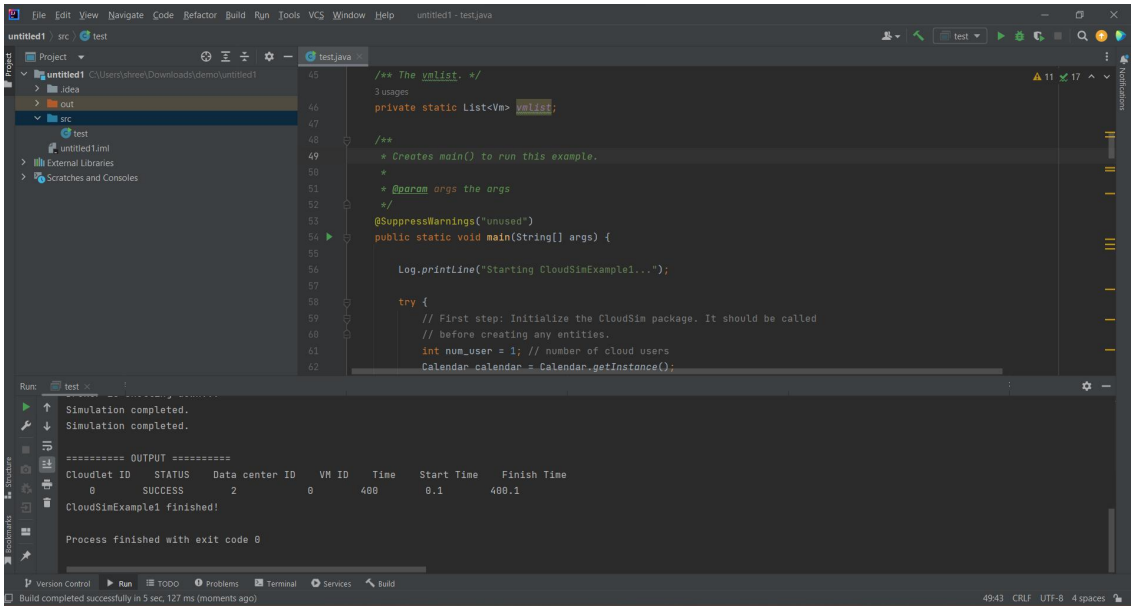


Step 4 Click Next and go to Libraries -> Add External JARs. Now browse to the same folder where you extracted your commons-math3 jar file and Open it.



Step 5 Finally click on Finish and wait for the project to build. After the project has been built, from the Project Explorer you can click on your project and from the dropdown go-to examples -> org.cloudbus.cloudsim.examples where you can find pre-written sample codes and try to run them.

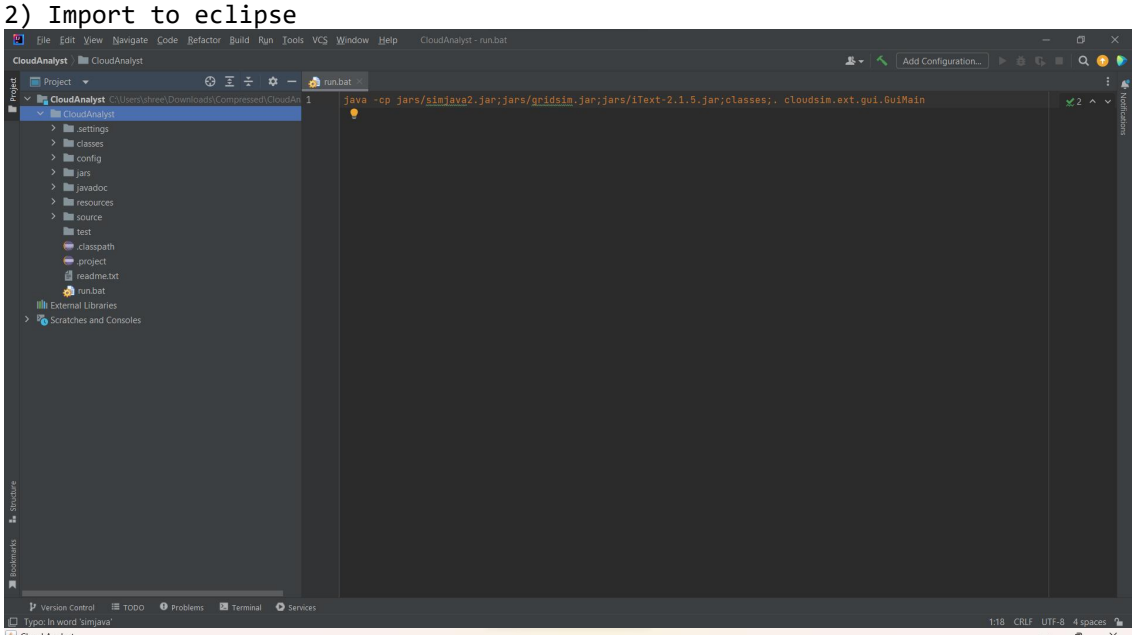

File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
untitled1 - test.java
Add Configuration...
Project
untitled1 C:\Users\shree\Downloads\demo\untitled1
.idea
src
test
untitled1.iml
External Libraries
Scratches and Consoles
test.java
import org.cloudbus.cloudsim.provisioners.RamProvisionerSimple;
/**
* A simple example showing how to create a datacenter with one host and run one
* cloudlet on it.
*/
public class test {
/** The cloudlet list. */
3 usages
private static List<Cloudlet> cloudletList;
/** The vmlist. */
3 usages
private static List<Vm> vmlist;
/**
* Creates main() to run this example.
*
* @param args the args
*/
@SuppressWarnings("unused")
public static void main(String[] args) {
Log.printLine("Starting CloudSimExample1...");
try {
// First step: Initialize the CloudSim package. It should be called
// before creating any entities.
int num_user = 1; // number of cloud users
Problems: Current File 28 Project Errors
Version Control TODO Problems Terminal Services
56:55 CRLF UTF-8 4 spaces

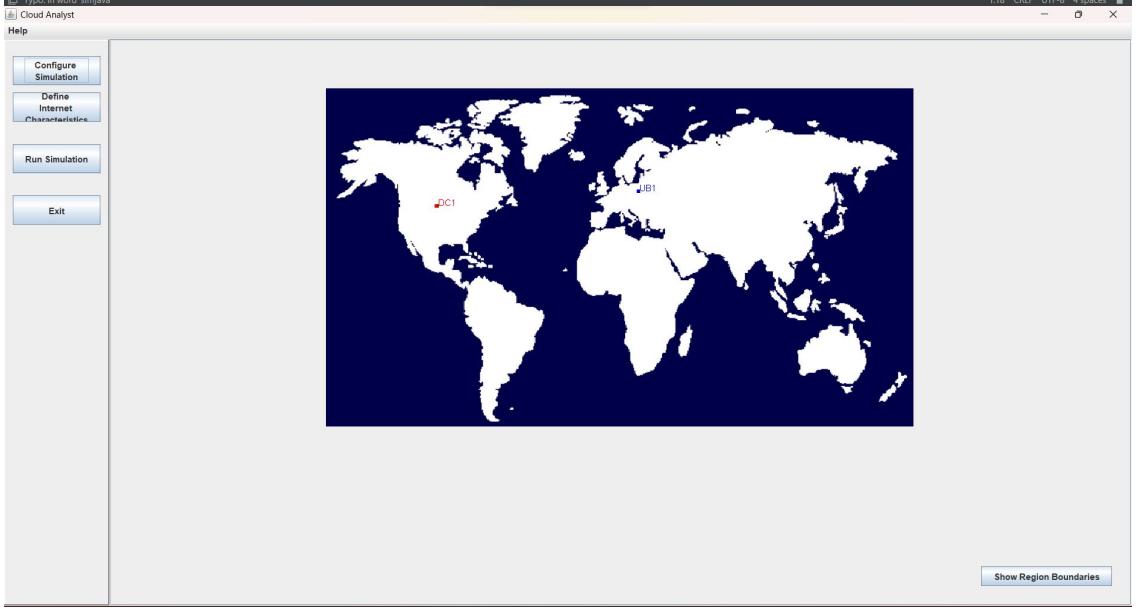

File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
untitled1 - test.java
test
Project
untitled1 C:\Users\shree\Downloads\demo\untitled1
.idea
out
src
test
untitled1.iml
External Libraries
Scratches and Consoles
test.java
/** The vmlist. */
3 usages
private static List<Vm> vmlist;
/**
* Creates main() to run this example.
*
* @param args the args
*/
@SuppressWarnings("unused")
public static void main(String[] args) {
Log.printLine("Starting CloudSimExample1...");
try {
// First step: Initialize the CloudSim package. It should be called
// before creating any entities.
int num_user = 1; // number of cloud users
Calendar calendar = Calendar.getInstance();
Run: test
Simulation completed.
Simulation completed.
========== OUTPUT ==========
Cloudlet ID STATUS Data center ID VM ID Time Start Time Finish Time
0 SUCCESS 2 0 400 0.1 400.1
CloudSimExample1 finished!
Process finished with exit code 0
Version Control Run TODO Problems Terminal Services Build
Build completed successfully in 5 sec, 127 ms (moments ago)
49:43 CRLF UTF-8 4 spaces

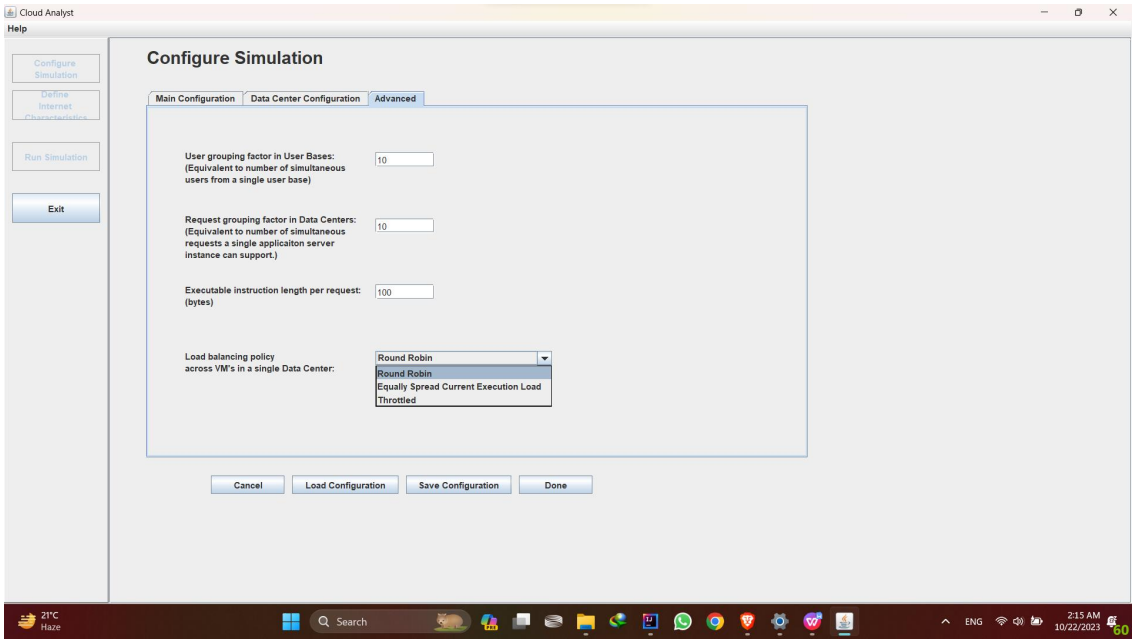
Part B

Cloud Analyst setup & Adding load balancing policy

1) Download cloud Analyst
2) Import to eclipse

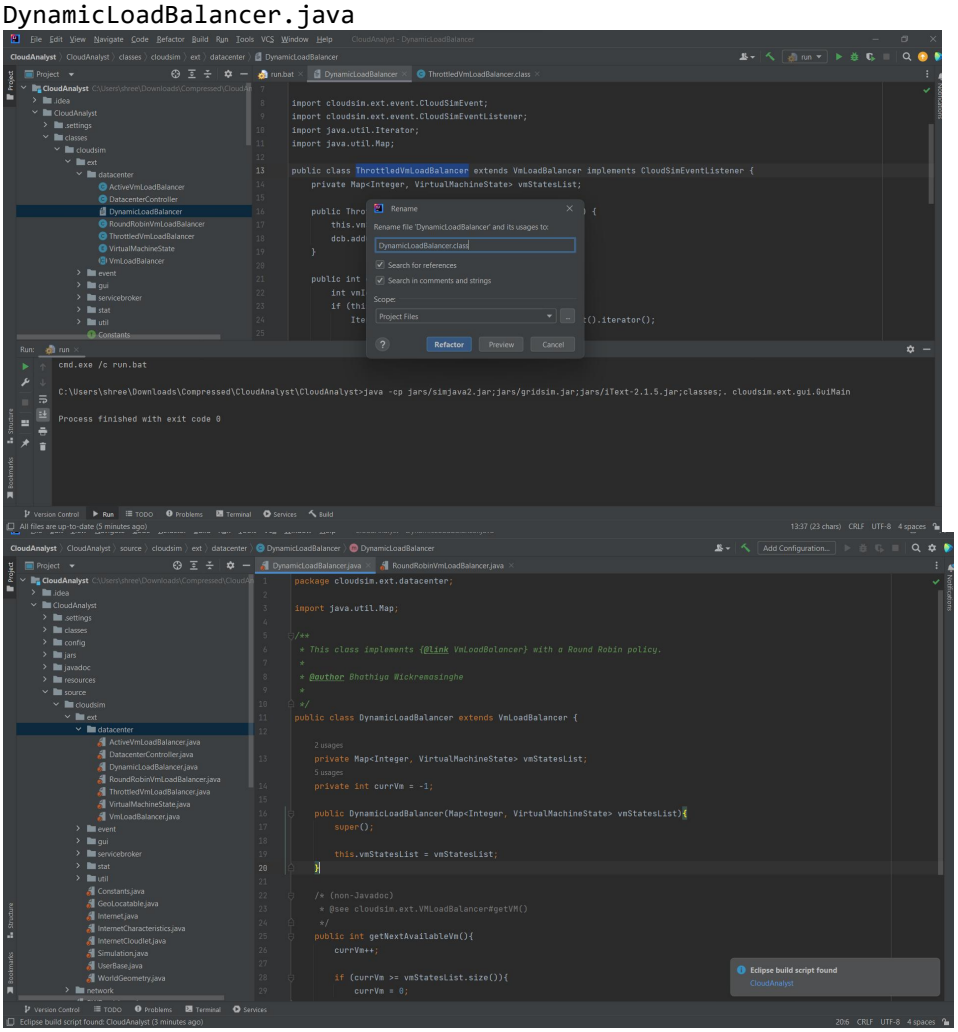




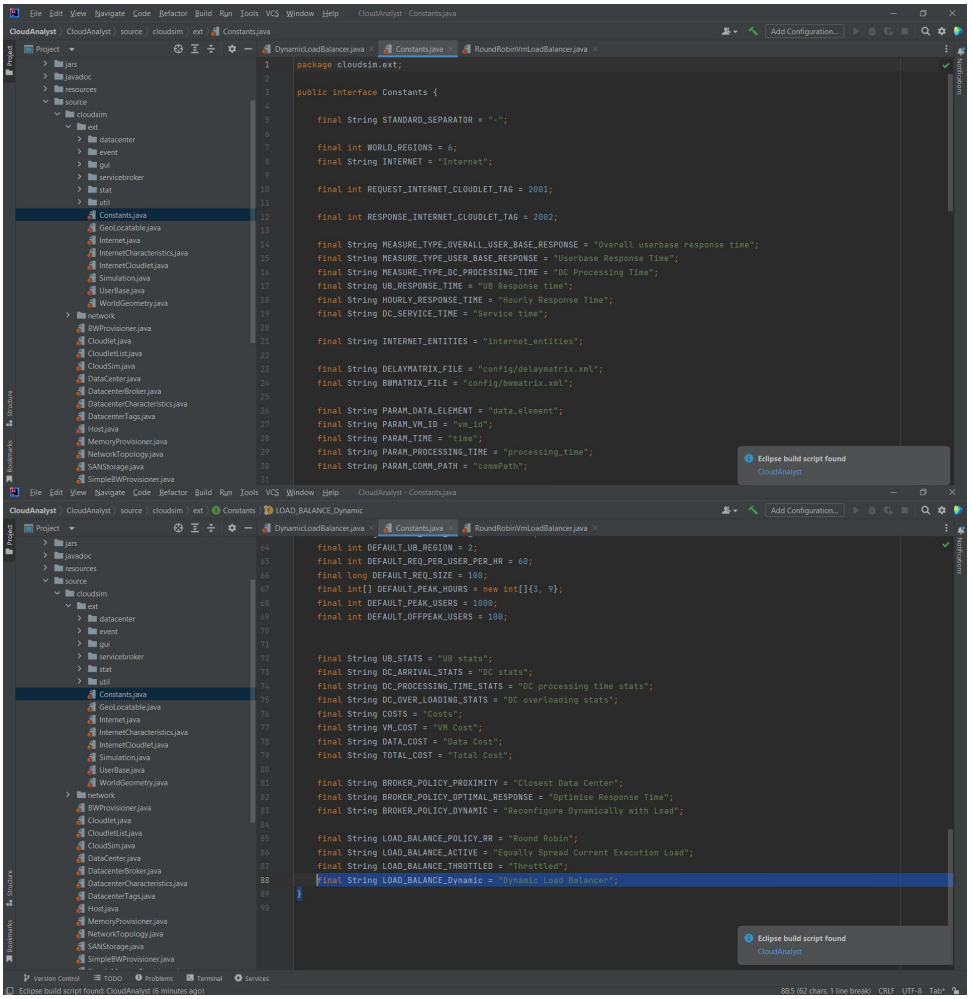


3) Add your algorithm

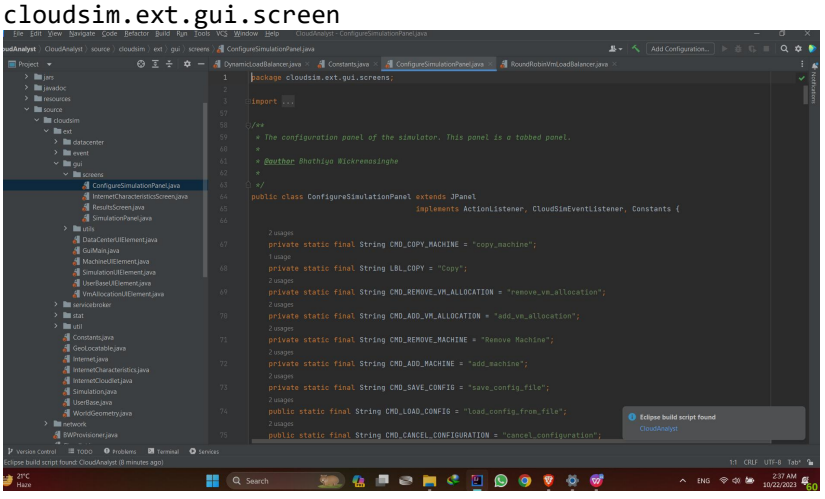
a) Create your own algorithm under cloudsim.ext.datacenter Call it DynamicLoadBalancer.java

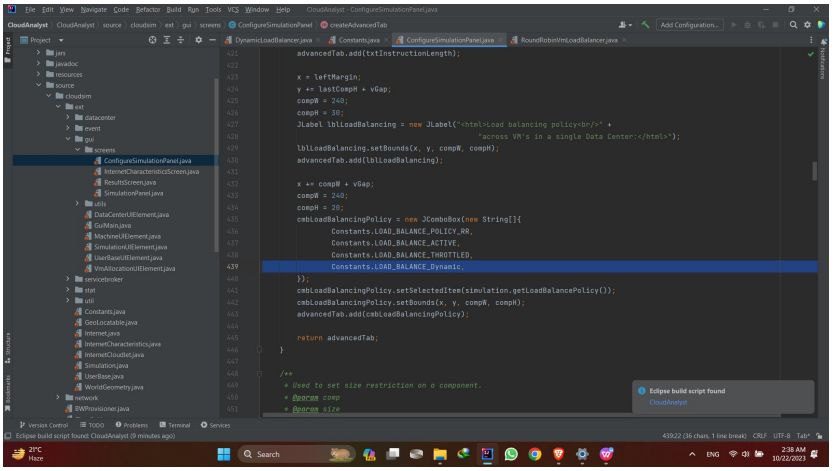


b) Create string in constant.java under cloudsim.ext Call it LOAD BALANCE_DLB

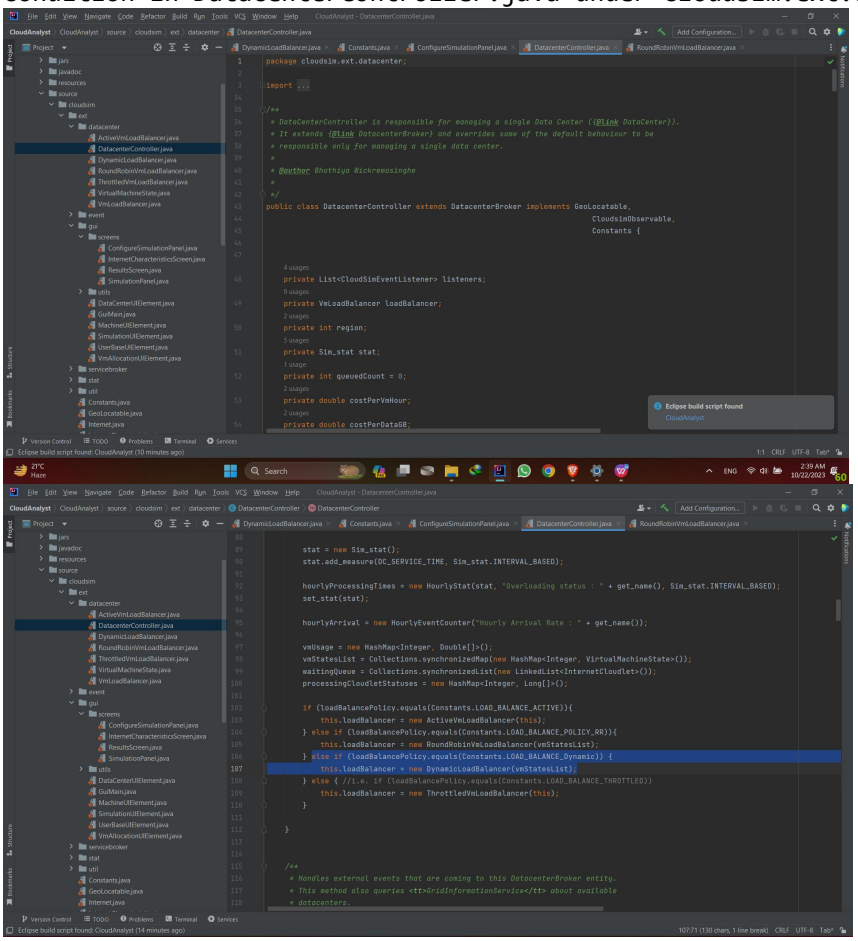


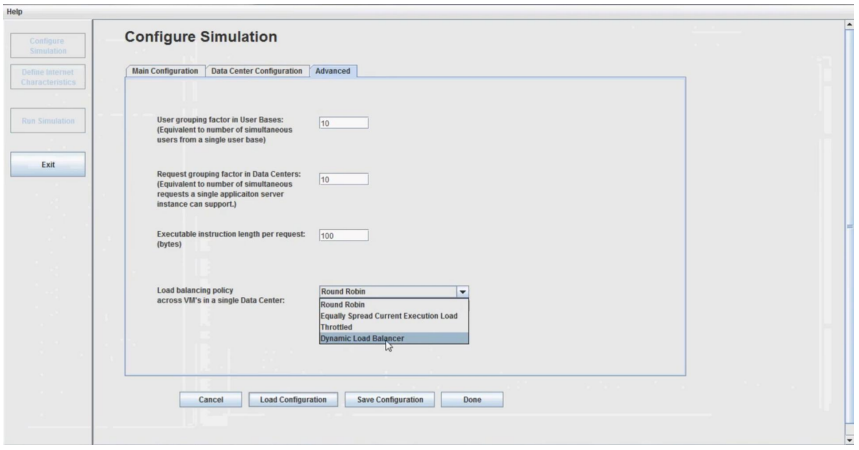
c) Add load balancing policy in ConfigureSimulationPanel.java under cloudsim.ext.gui.screen





d) Add load balancing policy (which you want to simulate) to ifelse condition in DatacenterController.java under cloudsim.ext.datacenter





ALL DONE.... NOW RUN