MANUAL BY
Adriel Kobe Nepomuceno
Dania Ashraf Mohammed

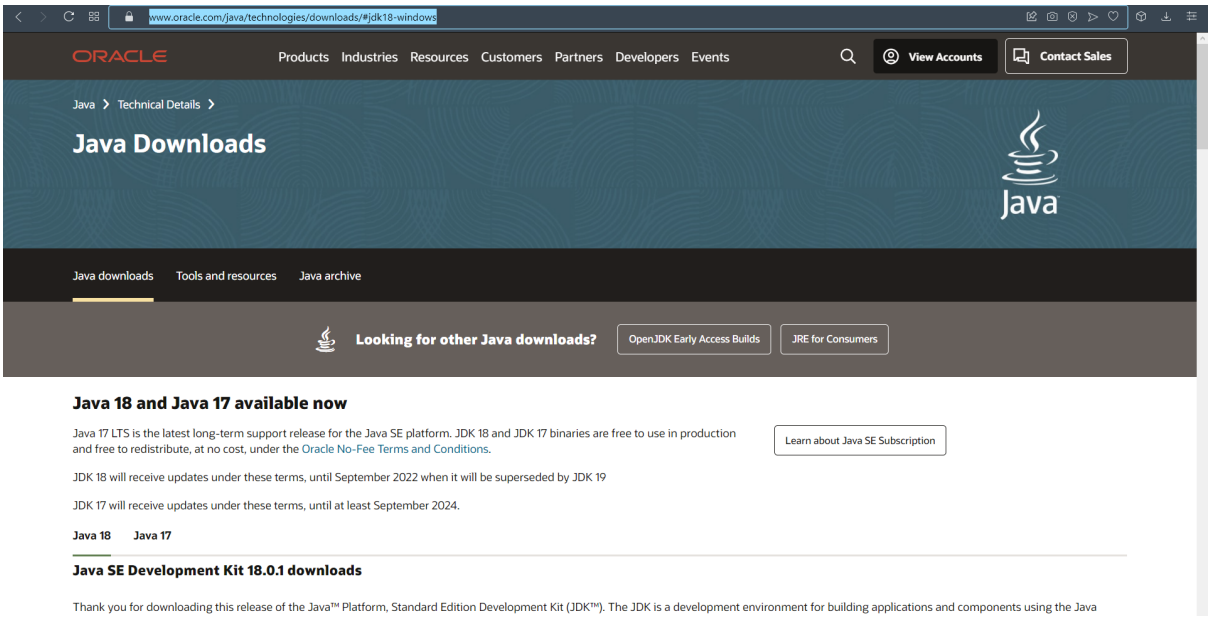
User Manual

Contents

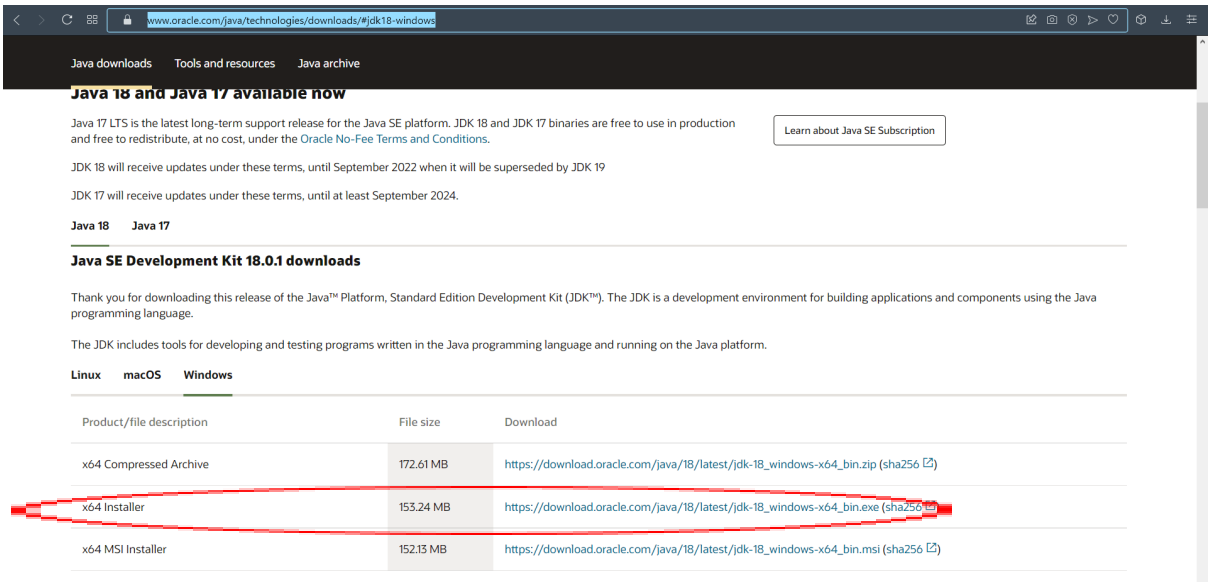
Prerequisites
Required Programs

In order to run the Quine-McCluskey Program, the user is required to have an installed copy of Java-14 or later. In order to install Java, follow the instructions below.

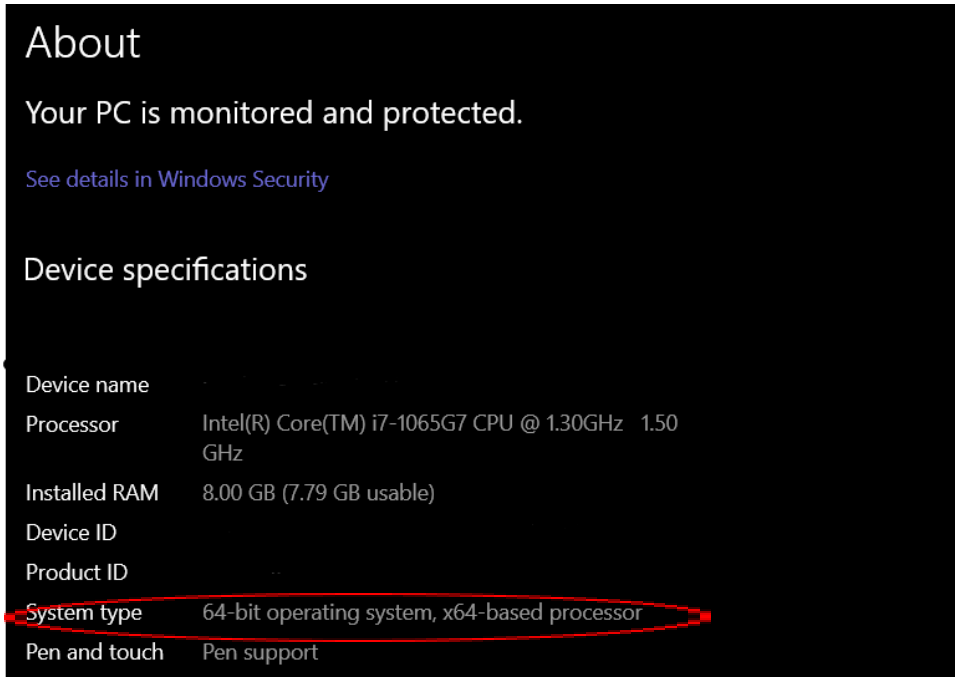
1. Go to the site: https://www.oracle.com/java/technologies/downloads/#jdk18-windows



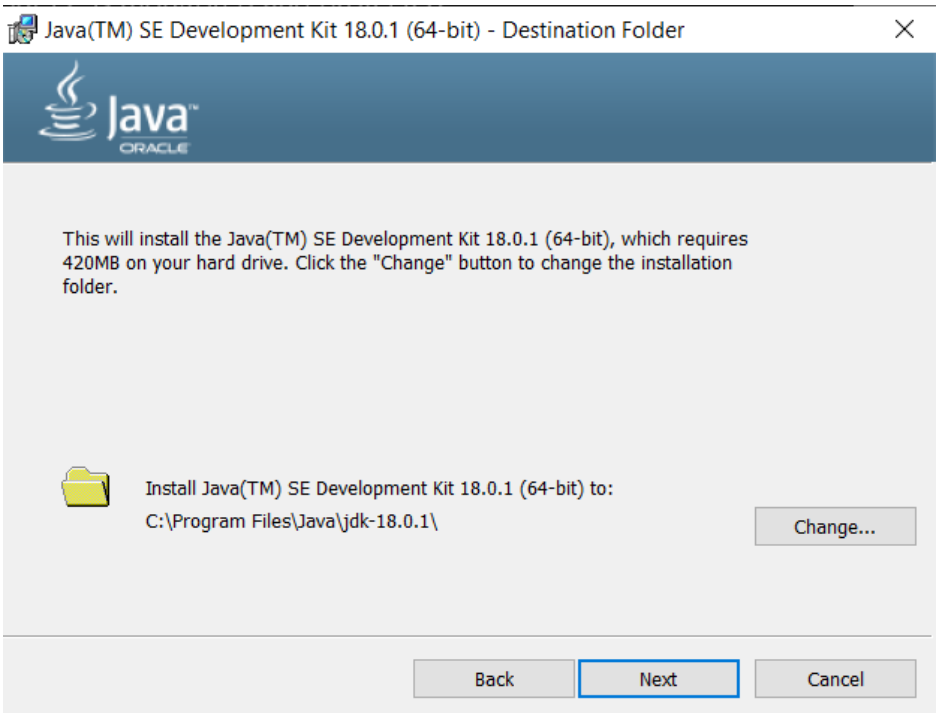
2. Scroll down and select the proper version for your system. You can choose the 'installer' type for easy installation.



For Windows users, make sure that your system is running on x64. To check go to Control Panel > System and there you can see your system components.



3. After downloading the installer, run it.

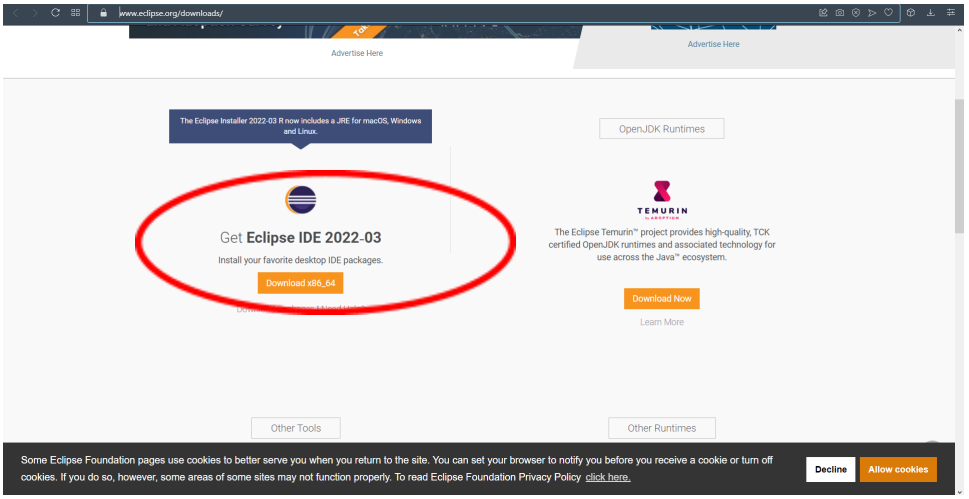


4. After running, wait for the installation to finish and you're done!

After installing Java, you would need an Integrated Development Environment (IDE) in order to run the program. We recommend using Eclipse IDE but you can use any IDE you want.

The instructions below are for installing Eclipse.

1. Go to the link https://www.eclipse.org/downloads/ and download the installer appropriate for your system. If you are using Windows, make sure your system is running on x64 or x86.



2. After downloading, run the installer and choose ‘Eclipse IDE for Java Developers’

3. Select your installation folder then click ‘Install’

4. After the installation you can now launch Eclipse.

System Requirements
In order to run Eclipse IDE properly, your system should meet the minimum recommended specifications.

Specs	Minimum	Recommended
Java Edition	1.4.0	5.0 or later
RAM Storage	512 MB	1GB or greater
Hard-Disk Space	300 MB	1GB or greater
Processor	800 Mhz	1.5 Ghz or faster

Rundown on the Quine-McCluskey Method

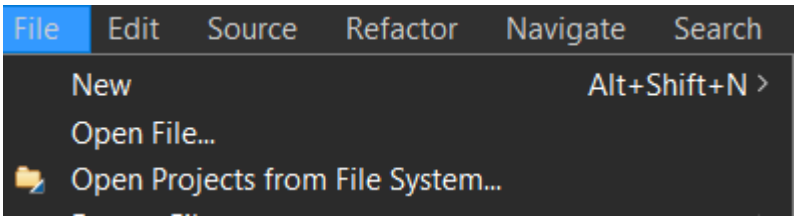
The presented program runs an implementation of the Quine-McCluskey Method. This method has its advantages over the K-map method in reducing boolean functions as it makes reducing functions with large amounts of variables easier. The method makes use of the functions prime implicants, makes comparisons between them, then creates the final reduced function. The following are the steps in doing the Quine-McCluskey Method.

1. Arrange the given minterms in ascending order and on their side, right their boolean value. Make sure that you fill in each variable in forming the boolean function
    a. If the function is f(a,b,c) = m(1,2), minterm 1 would be 001 and minterm 2 would be 010.
2. Group the minterms according to the amount of variables that turned up to be 1.
3. Compare each minterms from one group to the minterms of the succeeding groups. If two minterms' binary values only differ by 1 variable, they are considered a pair, add a pair to a new group and mark the minterms that were used in the pair with a check.
4. Repeat step 3 until all prime implicants are obtained.
5. List all implicants without a check mark onto the left side of a table. On the top, write each minterm that was given as part of the boolean function. Inspecting each prime implicant, put an 'X' on the minterms that are contained in the prime implicant.
6. Choose first the essential prime implicants, these are prime implicants that contain a variable unique to them. After that, continue choosing implicants in order to exhaust all minterms. The values of the selected prime implicants would be the reduction of the boolean function.
    a. If the final reduction for F(a,b,c) = m(1,2) would have both 1 and 2 as separate prime implicants whose values are 001 and 010 respectively, then F(a,b,c) = a'b'c + a'bc'

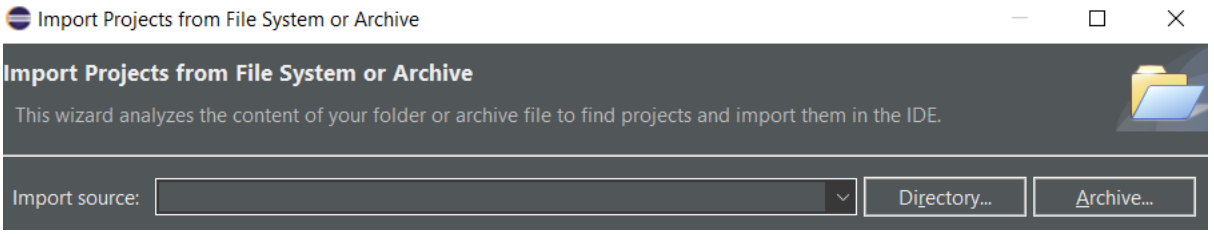
Understanding the Program
Navigating through Eclipse and Running Programs

The following are instructions in order to run the program.

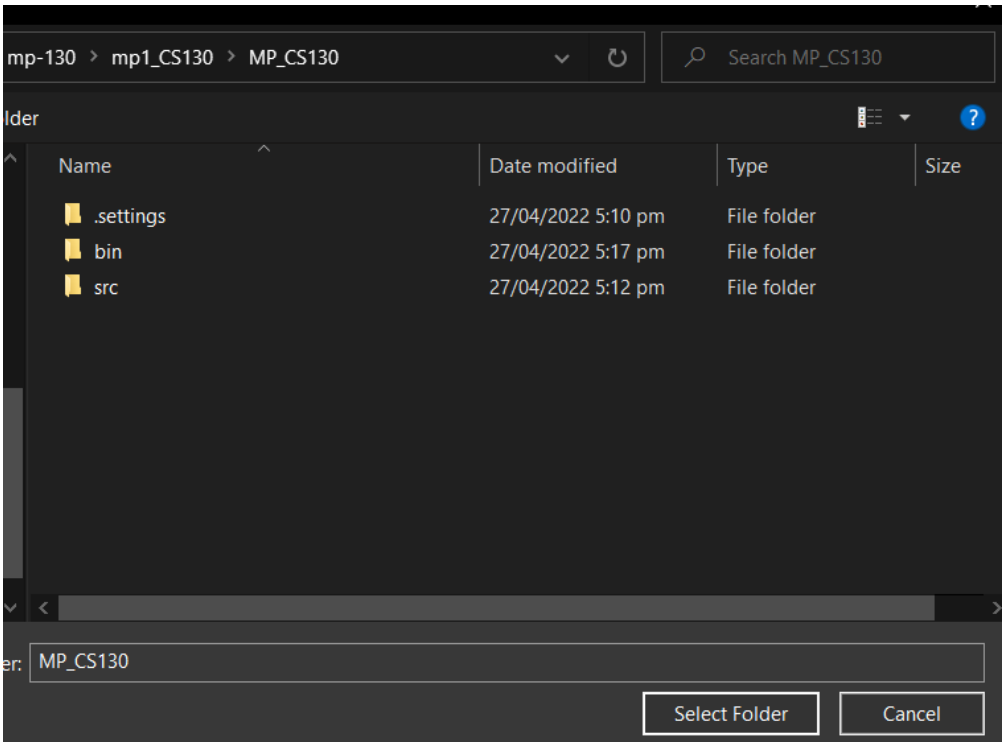
1. Launch Eclipse IDE.
2. In the file tab at the upper left corner of the screen, click ‘Open Projects from File System’



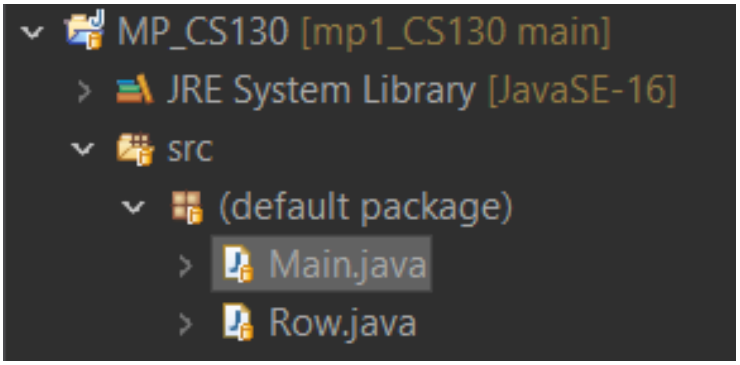
3. A new window will pop-up and from there, click “Directory” beside the “Import Source” bar



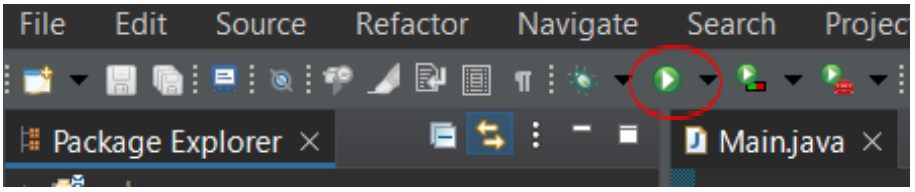
4. Navigate to the folder of the code. Make sure that you select the folder BEFORE the folder named “src”. Click “Select Folder”



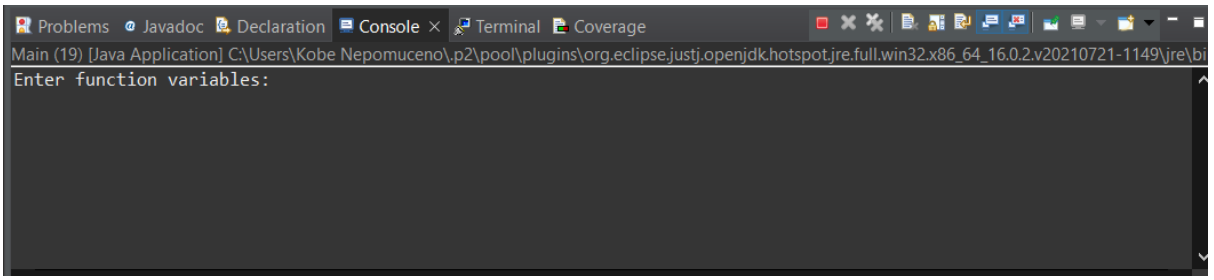
5. The project will be loading onto the side of the screen. Navigate to the folder “src” and inside it double-click on “Main.java”



6. On the top part of screen, click the green button with a play symbol to run the program.



7. The program will now run and produce outputs on the command line at the bottom of the screen.



Entering Inputs

Function Variables: The program will take in space-separated letter inputs (for example the string: “a b c d e” will be accepted.) The program will not take in any symbol that is not alphabetic (“& ! @ $” is not a valid string and is thus rejected by the program.) nor strings that contain more than one element joined before a space (“a bc d ef” is not a valid string). Numerical values will also not be accepted as variables.

Minterms: The program will take in space-separated numerical values for minterms only provided that they are less than or equal to the maximum value that can be stored in the entered variables. (“a b c d e” has 5 variables meaning it can take values from 0 - 31).

Understanding the Outputs

An example output is presented below

```
Enter function variables: a b c d e
Enter minterms: 1 2 3 4 5 10 11 17 19 20
{11=01011, 1=00001, 2=00010, 3=00011, 4=00100, 5=00101, 17=10001, 19=10011, 20=10100, 10=01010}
1       00001   1
2       00010   1
4       00100   1
3       00011   2
5       00101   2
17      10001   2
20      10100   2
10      01010   2
11      01011   3
19      10011   3

COMPARISON: 1
-----------------------------------------------------
1,3     000-1   1
1,5     00-01   1
1,17    -0001   1
2,3     0001-   1
2,10    0-010   1
4,5     0010-   1
4,20    -0100   1
3,11    0-011   2
3,19    -0011   2
17,19   100-1   2
10,11   0101-   2

COMPARISON: 2
-----------------------------------------------------
2,3,10,11       0-01-   1
1,3,17,19       -00-1   1
1,3,17,19       -00-1   1
2,3,10,11       0-01-   1
1,5     00-01   1
4,5     0010-   1
4,20    -0100   1
```

```
PRIME IMPLICANT TABLE
-----------------------------------------------------------
              1    2    3    4    5    10   11   17   19   20
2,3,10,11          X    X              X    X
1,3,17,19     X         X                        X    X
1,5           X                   X
4,5                            X    X
4,20                           X                             X

-----------------------------------------------------------
Output: a'c'd+b'c'e+b'cd'e'+a'b'cd'
```

Each comparison stage is arranged in the form of minterm, Binary representation, group, respectively. Note that there will be some unequal spacing in some instances as strings are now long and the compiler has to compensate for its length. Each stage shows prime implicants that have been compared and paired together up to that stage.

The prime implicant table presents the unused prime implicants along with the minterms that are contained in them. Internally, they are checked for the essential prime implicant, if none are found, they instead take the one that has the largest amount of unused minterms contained in them.

The output gives the final reduced boolean function based on the method.

Troubleshooting and Frequently Asked Questions

"I cannot find the file that I downloaded."

Check your downloads folder or your desktop as these are the places that browsers commonly save downloaded files. If it isn't there, check your browser's settings on where it saves the downloaded files.

"I can't install Java"

You might be running on a non-compatible system. Make sure that your system is running at at least x64. If it is and it is still not working, you may need to uninstall Java then reinstall it again. You can head to Java's troubleshooting page for more information: https://docs.oracle.com/javase/10/troubleshoot/general-java-troubleshooting.htm

"Is installing these programs safe?"

They are 100% safe to install as long as you follow the links that are provided in this document. Remember to not download from illegal sites and other unofficial sites.

"The output of the program isn't aligned. How do I fix this?"

The program outputs data based on a hard coded template. The strings for minterms tend to get longer with each passing comparison stage and so, the compiler compensates for this length by forcefully shifting each string after it. You may fix it if you know how to code in the Java language.

"How many variables can I use?"

You can use any amount of variables you need as long as they are alphabetic. Note that the system takes lowercase and uppercase characters as different variables so the variable "a" is considered different from the variable "A" and thus can be used together.

Repository

You may access the repository of this code along with other deliverables such as a java doc on its implementation at: https://github.com/KobeNepomuceno/mp1_CS130