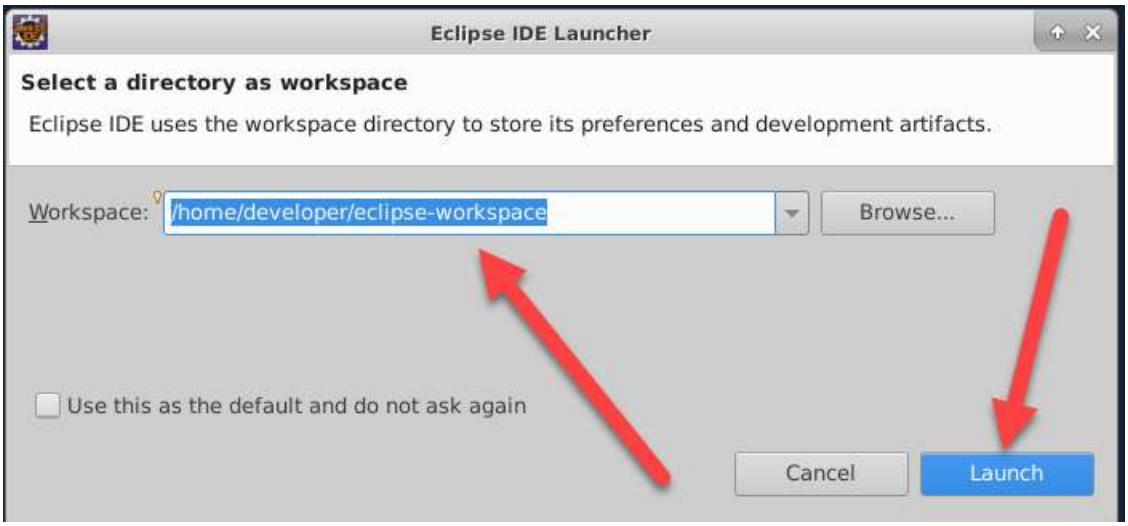
Start the lab

Lab2: Using the Scanner class for Console applications

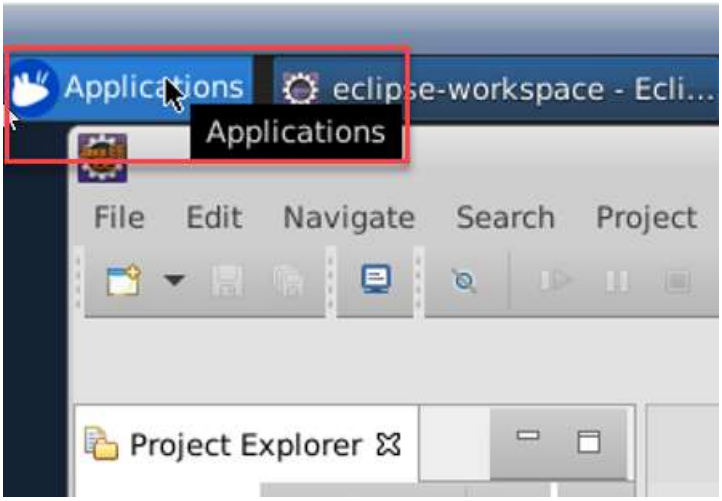
Launch Lab 2

Let Eclipse start

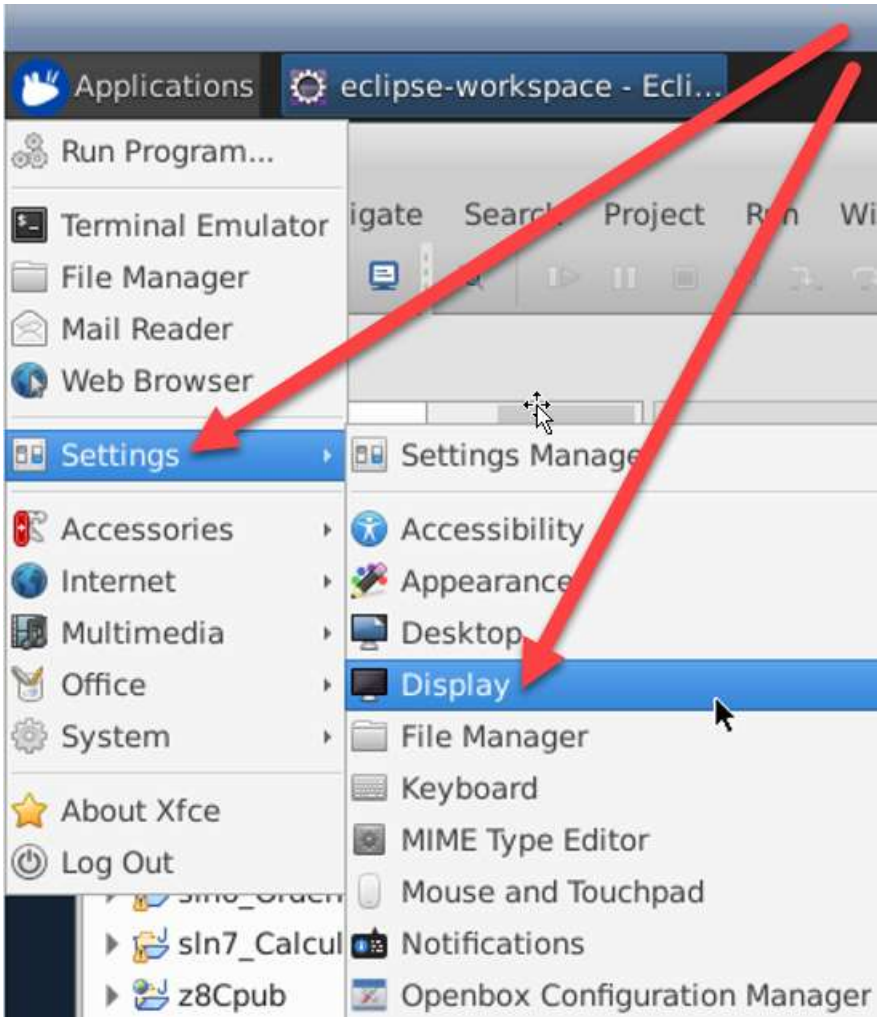
Leave the workspace unchanged, and click **Launch.**



In the upper left hand corner, click the **Applications** button.

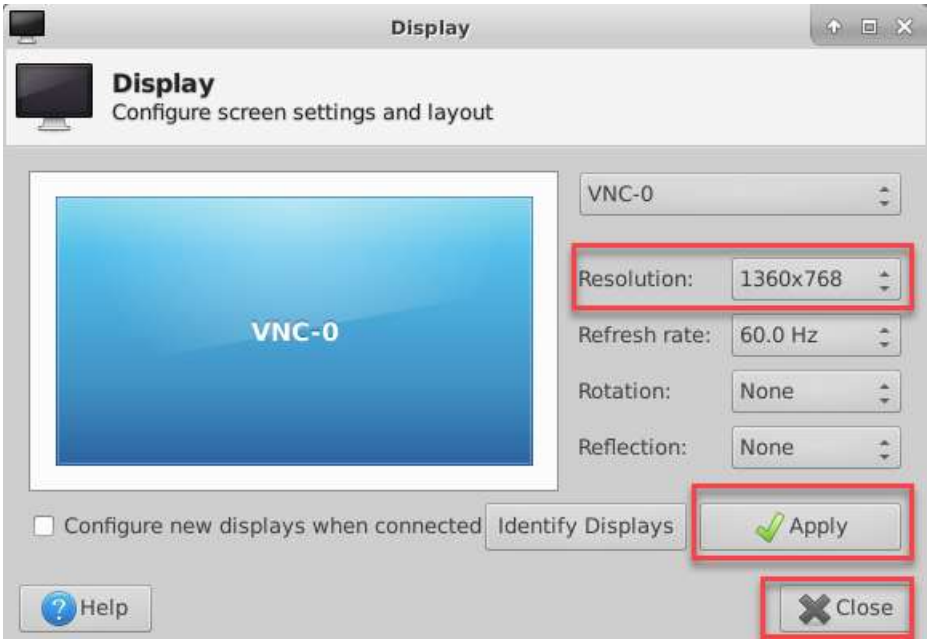


Click **Settings** -> **Display**

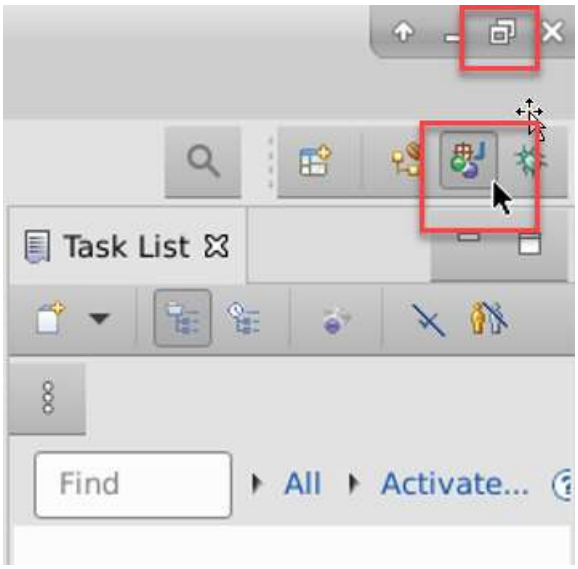


IMPORTANT: This example uses resolution **1360x768**. However, a different setting may work better for you.

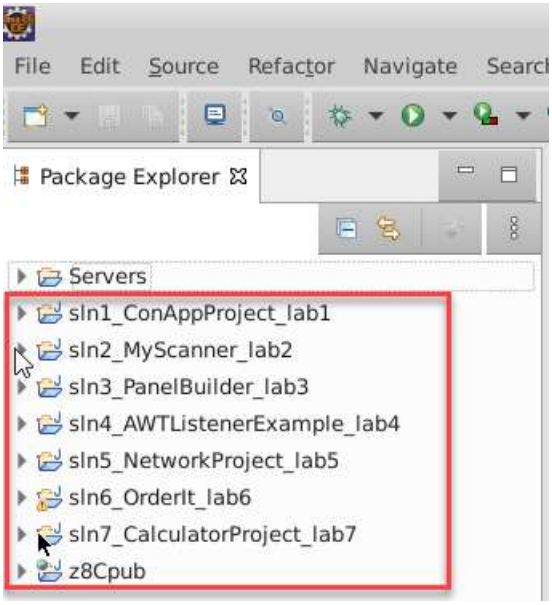
Click **Apply** and **Close**.



In the upper right corner, maximize **Eclipse**. Also, important, click the **Java Perspective button**. It must be the **Java perspective**.



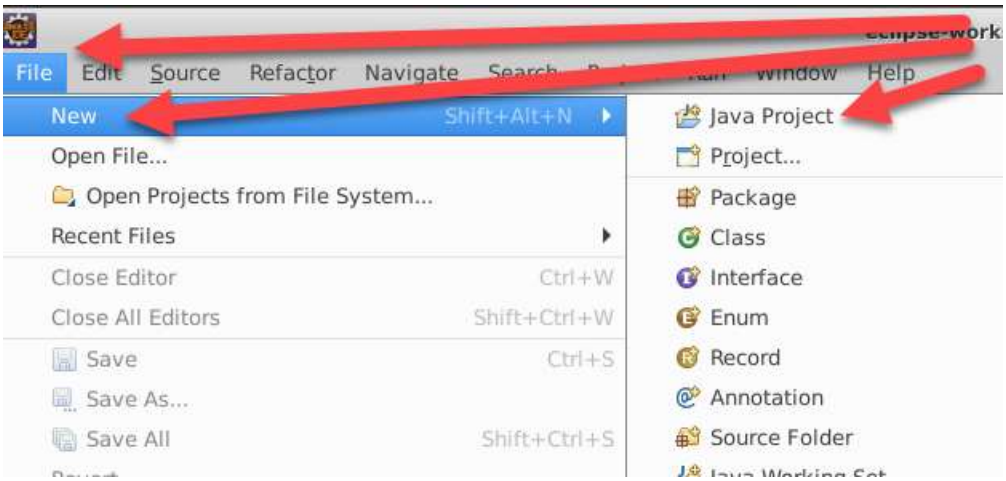
Solution projects are listed in the Project Explorer. Key in all lines of code. However, if needed, use the solutions for troubleshooting.



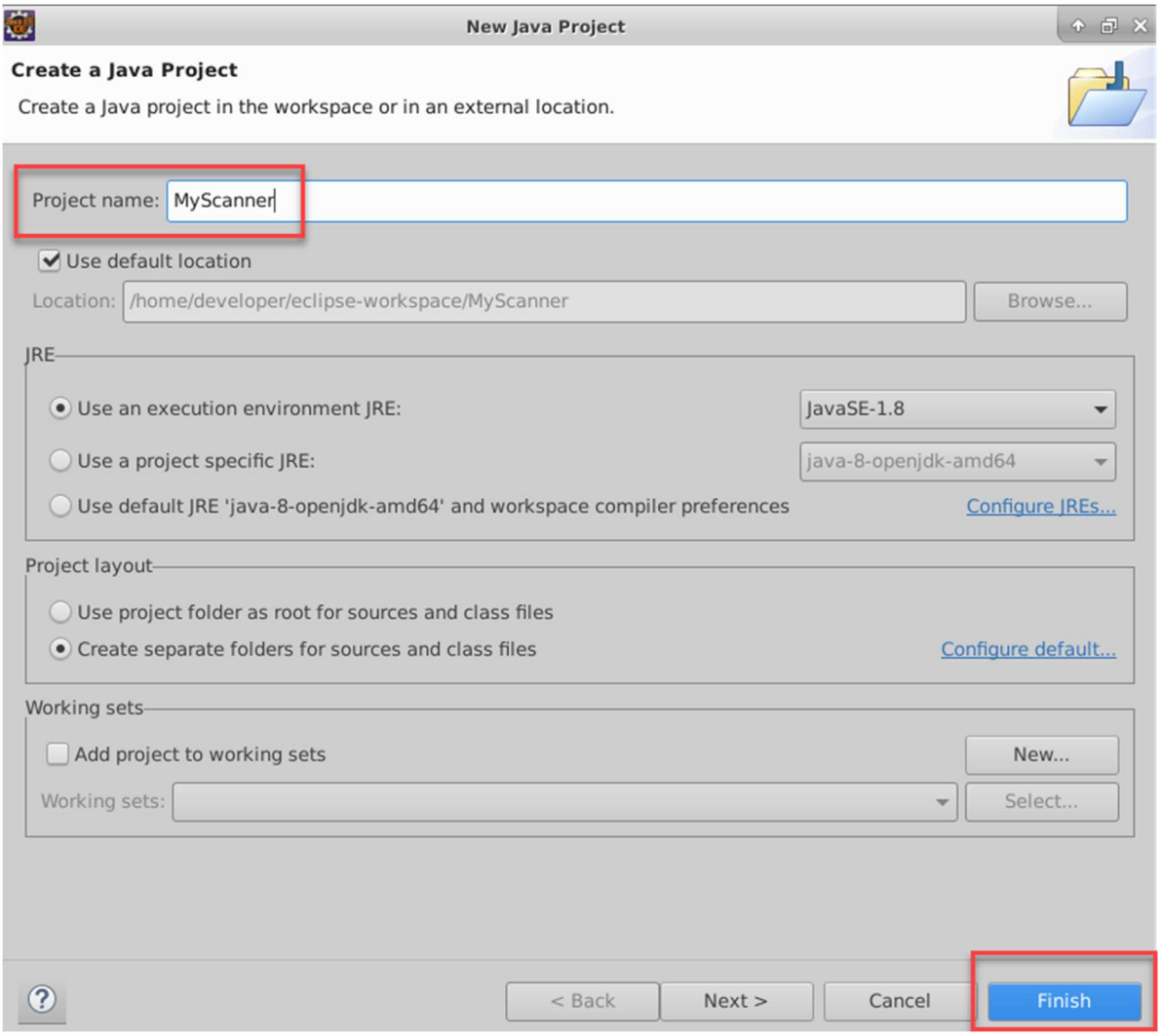
For this first lab, use solution project **sln2_MyScanner_lab2** for reference. It is possible to copy and paste the Java code. However, avoid copy and paste as much as possible.

Create a new Java Project called **MyScanner**

Click the **File** menu -> **New** -> **Java Project**

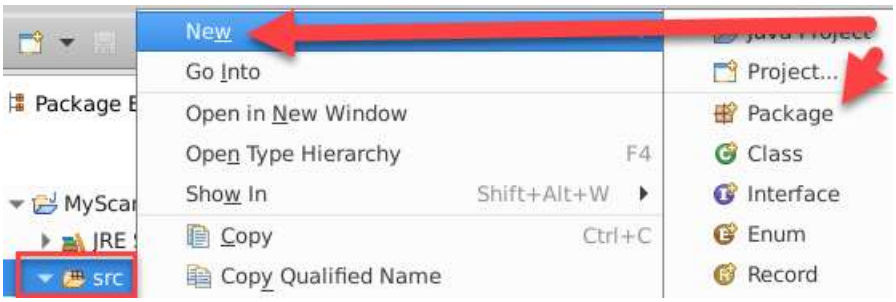


Name the project **MyScanner** and click **Finish**.

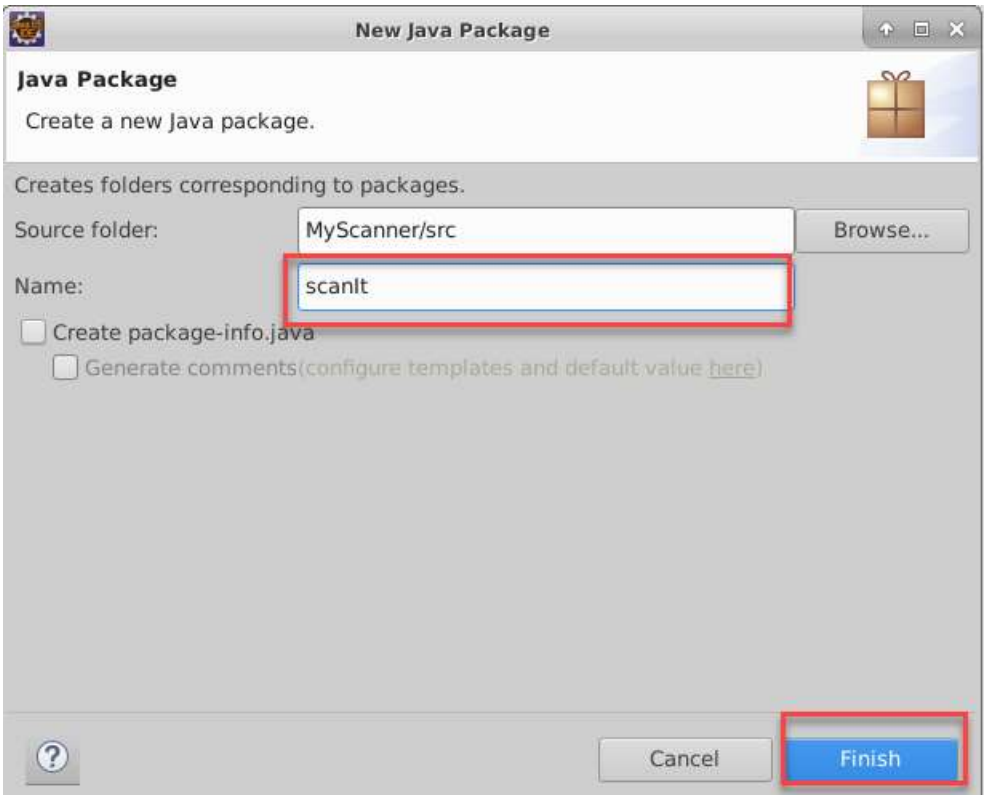


Create a package called addIt

Expand the **MyScanner** and right click on the **src** folder, select **New** -> **Package**

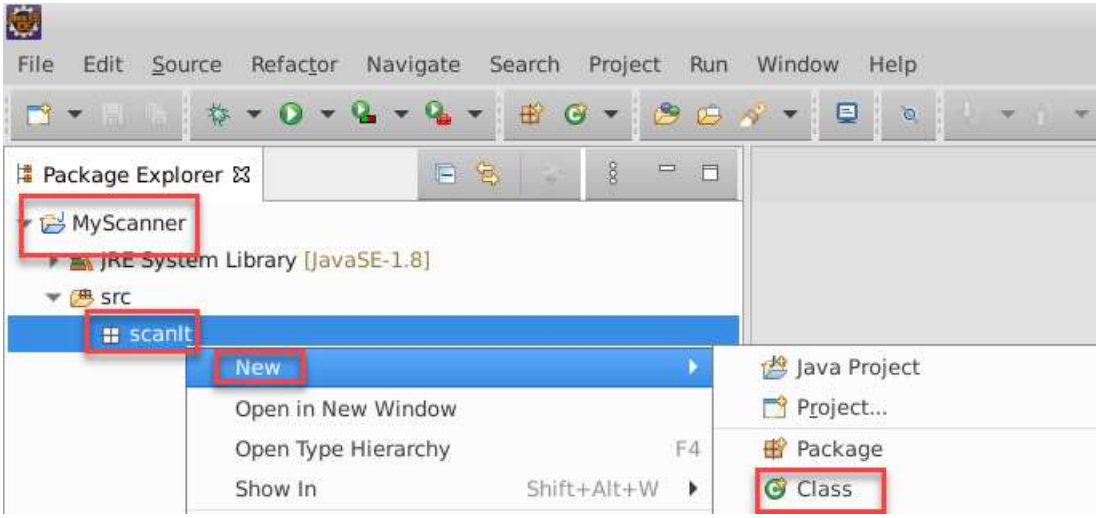


When the Java Package dialog opens, enter:
Name: **scanIt**



Click **Finish**

Right click on the **scanIt** package and select, New -> Class

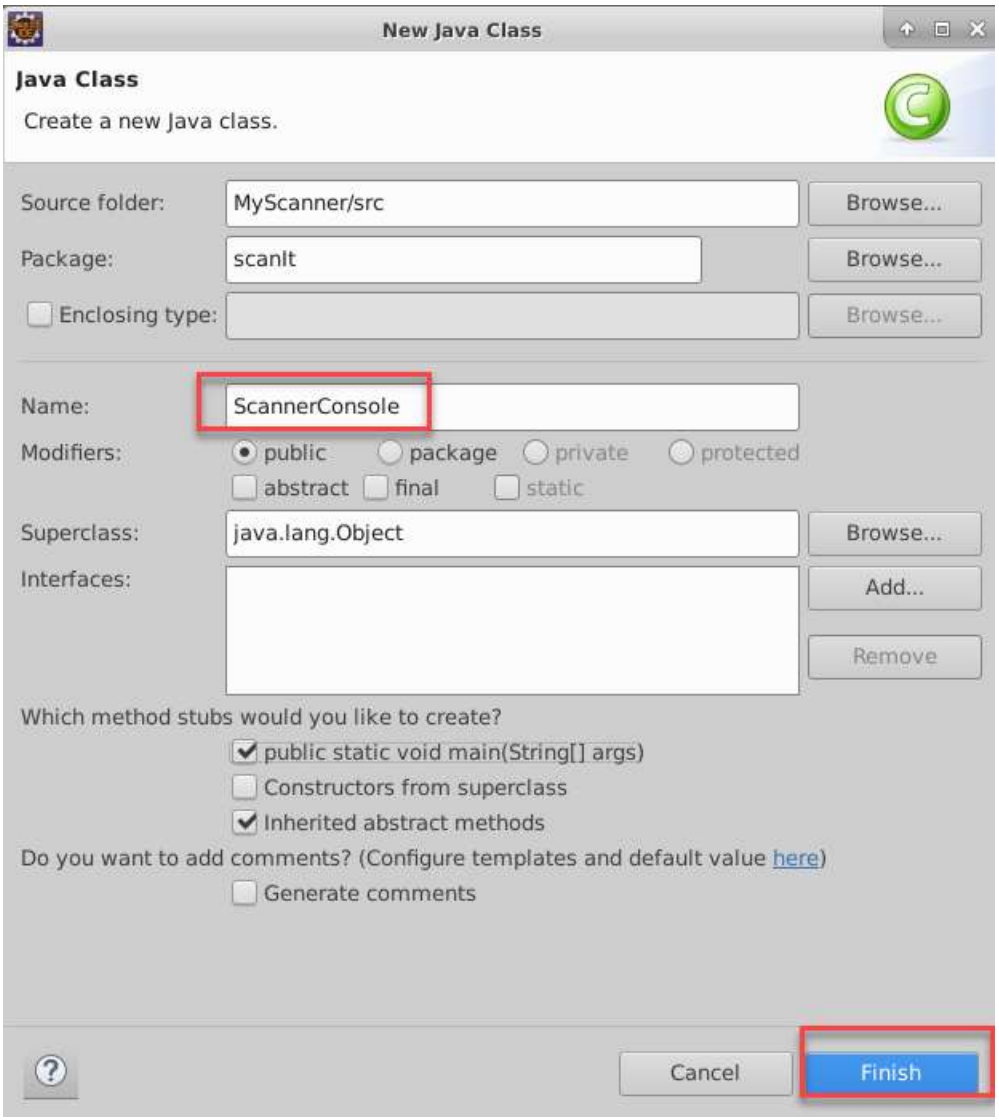


In the Java Class dialog, enter the following:

Name: **ScannerConsole**

Check: **public static void main(String[] args)**

Click: **Finish**



Eclipse will open the **ScannerConsole.java** class file.

```
ScannerConsole.java
1  package scanIt;
2
3  public class ScannerConsole {
4
5      public static void main(String[] args) {
6          // TODO Auto-generated method stub
7
8      }
9
10 }
11
```

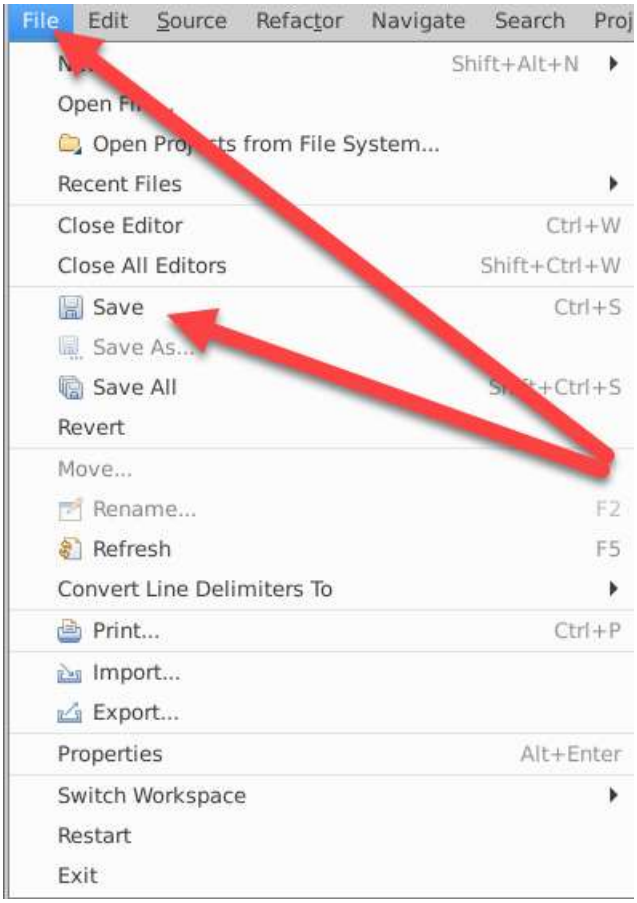
Add the library that support command line input. It is just 1 library.

import java.util.Scanner;

These libraries support command line input, and converting the input to data.

```
ScannerConsole.java
1  package scanIt;
2
3  import java.util.Scanner;
4
5  public class ScannerConsole {
6
7      public static void main(String[] args) {
8          // TODO Auto-generated method stub
9
10     }
11
12 }
```

Click the **File** menu -> **Save**



It is time to instantiate class. We do so in the main method.

In the main method add the code below

Notice there is no **throws IOException**

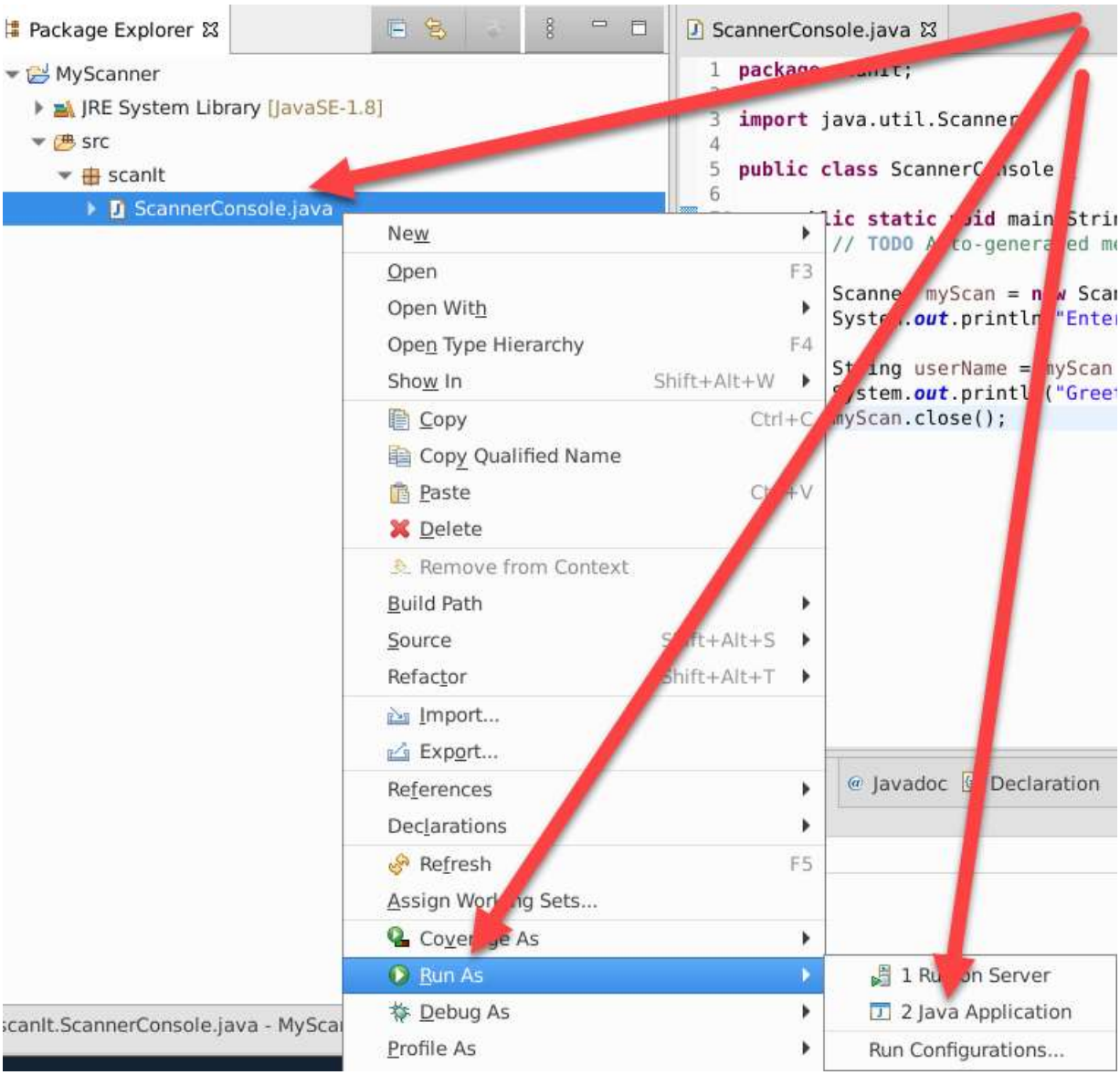
ScannerConsole.java

```
1  package scanIt;
2
3  import java.util.Scanner;
4
5  public class ScannerConsole {
6
7      public static void main(String[] args) {
8          // TODO Auto-generated method stub
9
10         Scanner myScan = new Scanner(System.in); //Instantiate myScan as a Scanner object
11         System.out.println("Enter username");    // Instruction to the user output to the console
12
13         String userName = myScan.nextLine();         //take command line input and store it in a String
14         System.out.println("Greetings " + userName);   //output the results to the console.
15         myScan.close();                                //Delete the myScan object to free the memory.
16
17     }
18
19 }
20
```

Click **File** menu -> **Save**

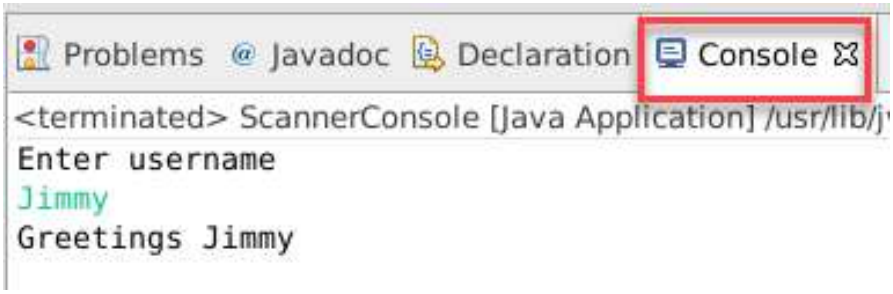
Run the application

In the Package Explorer view, right click the ScannerConsole.java file, click **Run As** -> **Java Application**



Click in the **Console** view, to focus it.

Enter a name from the keyboard



The Scanner methods below are available for different data. Therefore, there is less of a need to convert data.

nextInt(), returns int data.

nextDouble(), returns double, or real number data.

Update the application by adding all the lines shown below.

Add the getTotal method shown below. It returns double data.

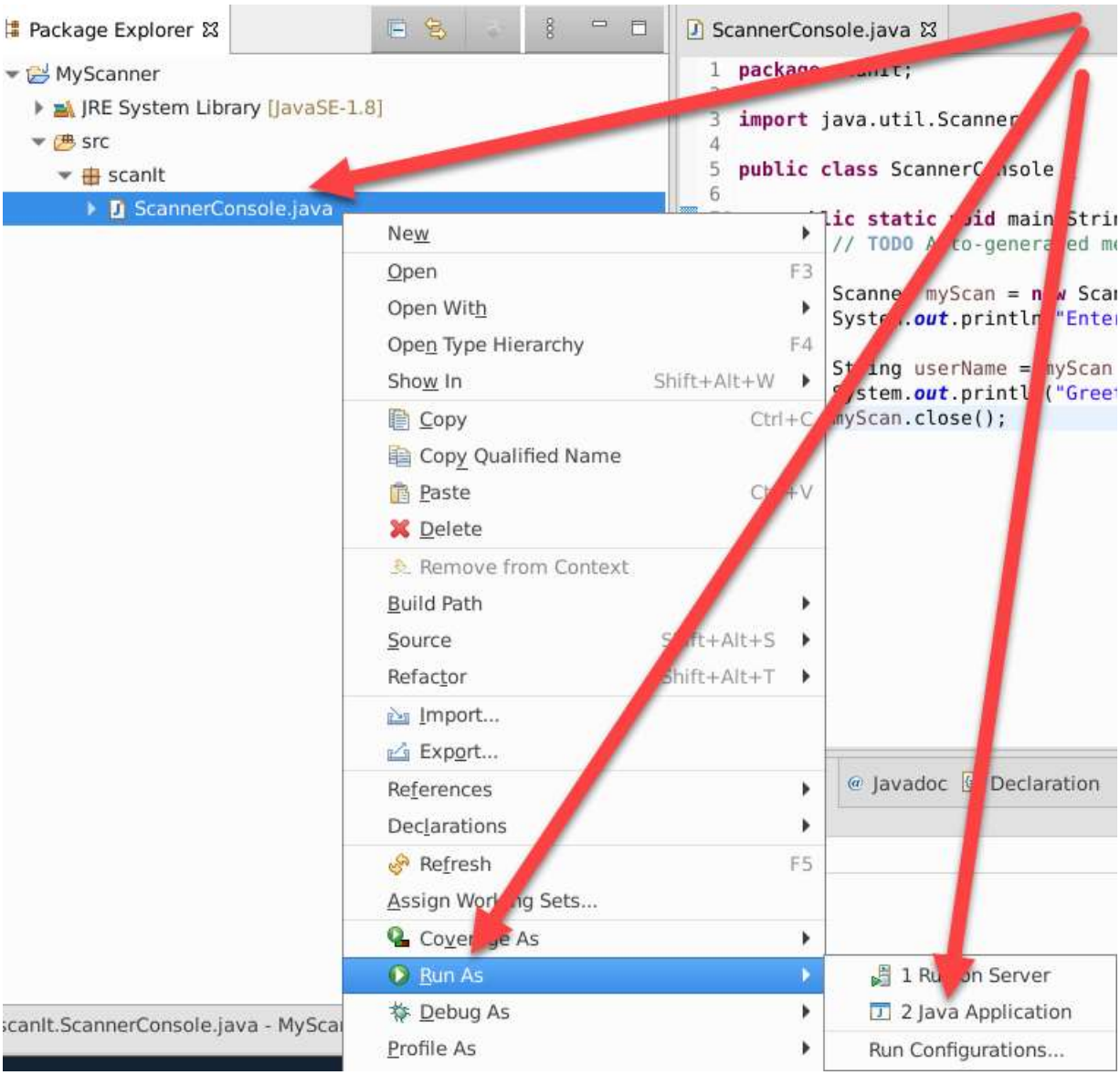
Add the additional code shown, below, in the main method. The code is command interaction with the user.

ScannerConsole.java

```java
package scanIt;

import java.util.Scanner;

public class ScannerConsole {

    public double getTotal(double cst, int amt) {

        double cost = cst;
        int num = amt;
        double tot = cost * num;

        return tot;
    }

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        ScannerConsole sc = new ScannerConsole(); //instantiate ScannerConsole class and create object sc

        Scanner myScan = new Scanner(System.in); //Create object myScan of type Scanner. Get command line input
        System.out.println("Enter Username: "); //Instructing output to the user
        String userName = myScan.nextLine(); //call method nextline on myScan. It returns a String.

        System.out.println("Enter the number of items purchased: "); //Instructing output to the user
        int items = myScan.nextInt(); //call method nextint on myScan. It returns int data.

        System.out.println("Enter the price of the items purchased: ");  //Instructing output to the user
        double price = myScan.nextDouble();   //call method nextdouble on myScan. It returns double data.

        double total = sc.getTotal(price, items); //call getTotal on sc object which is of type ScannerConsole.

        System.out.println("Greetings: " + userName);           //Output username to the console
        System.out.println("The number of items is: " + items); //Output items to the console
        System.out.println("The price is: " + price);           //Output the price to the console
        System.out.println("The total is: " + total);           //Output the total to the console
        myScan.close();   //delete the myScan object from memory.
    }
}
```

Run the application

In the Package Explorer view, right click the ScannerConsole.java file, click **Run As** -> **Java Application**

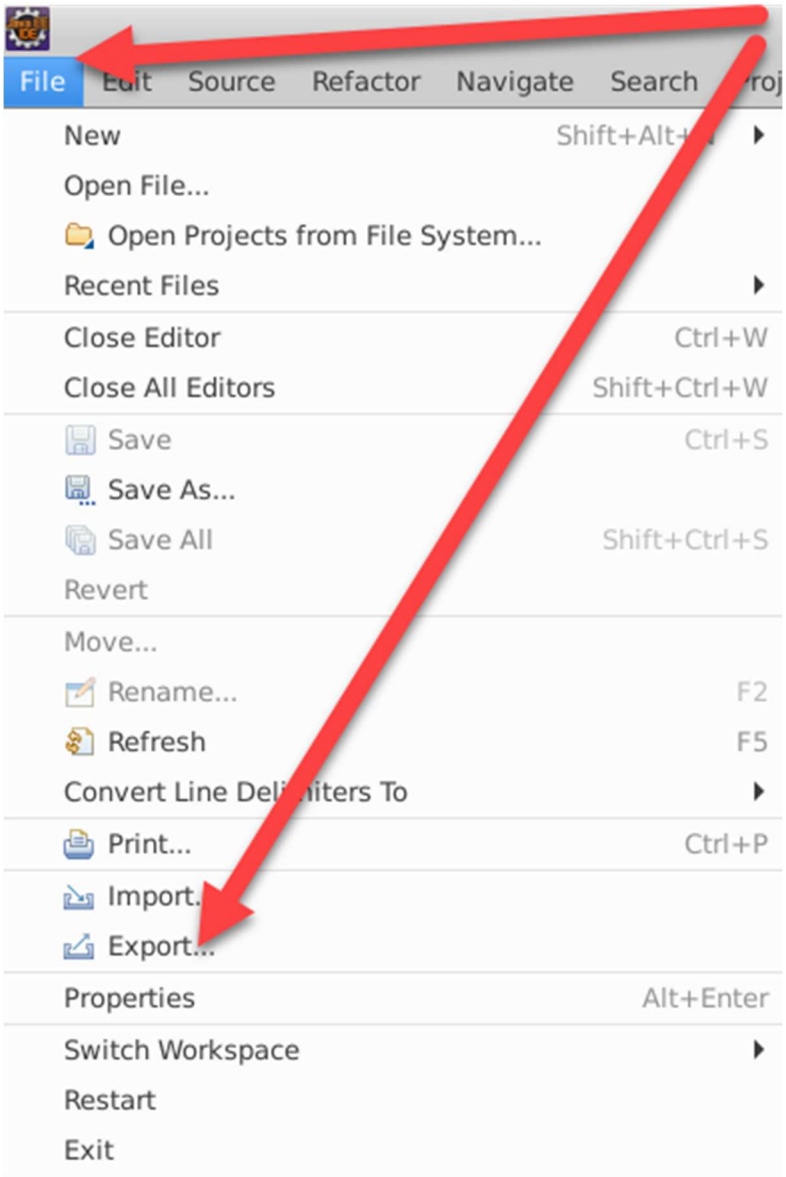


Problems @ Javadoc Declaration Console

```
<terminated> ScannerConsole [Java Application] /usr/lib/j
Enter Username:
Jimmy
Enter the number of items purchased:
8
Enter the price of the items purchased:
18.43
Greetings: Jimmy
The number of items is: 8
The price is: 18.43
The total is: 147.44
```

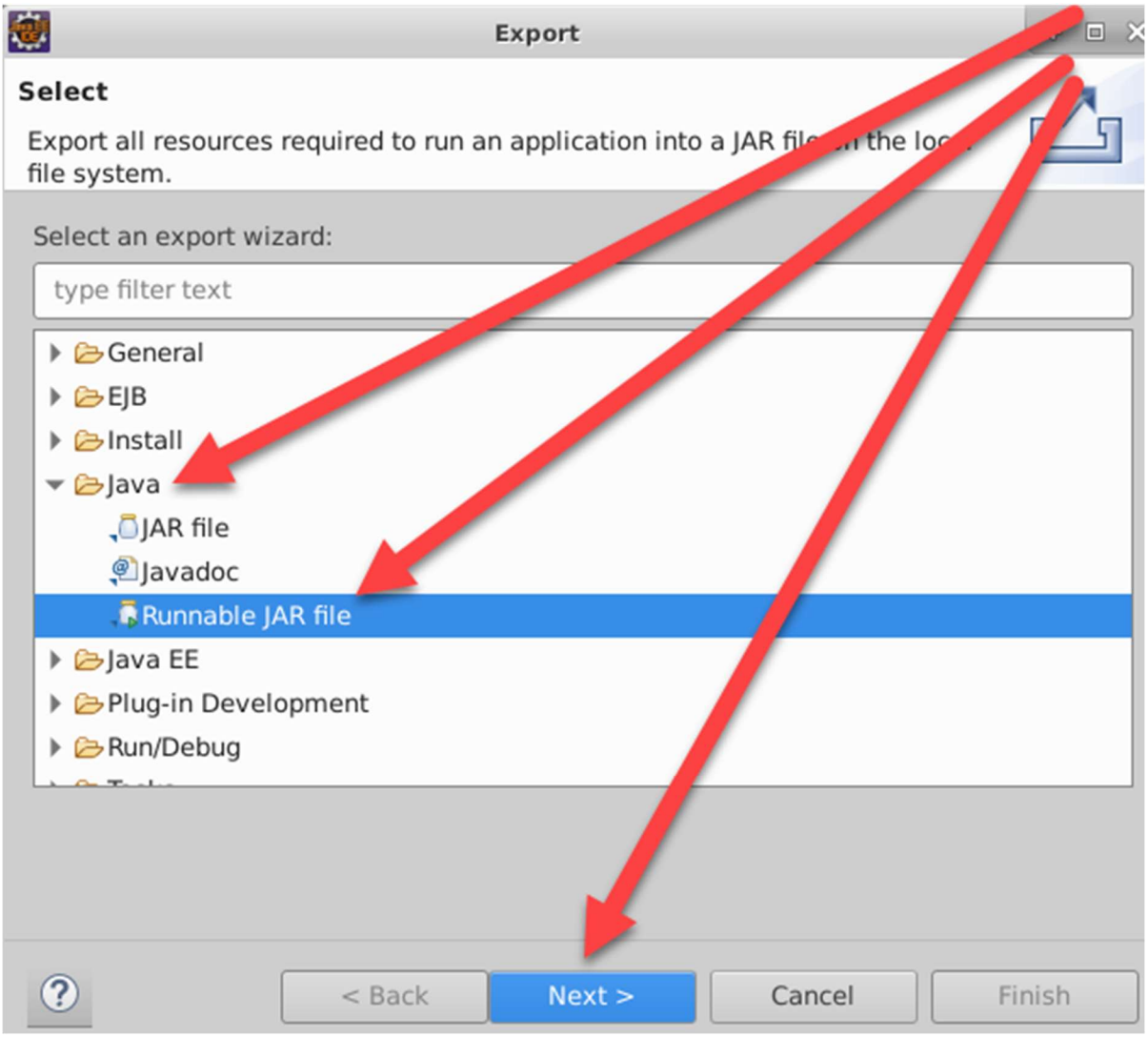
Export a jar file to run from the command line

Click **File** menu -> **Export**



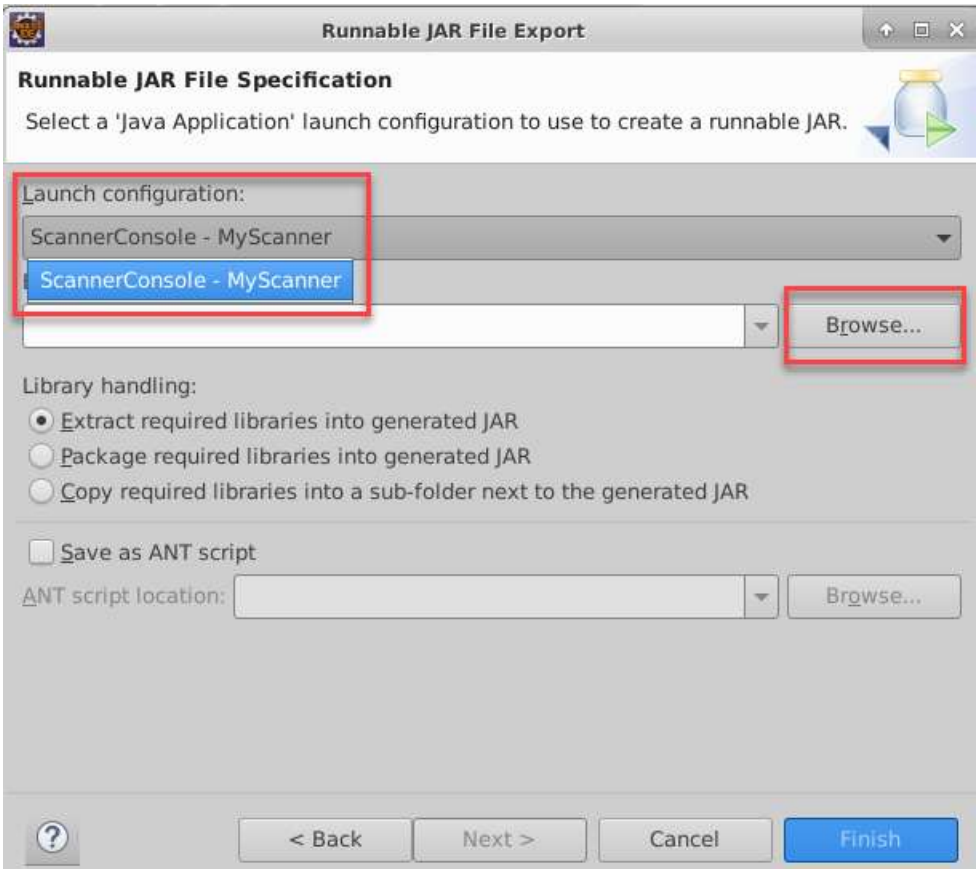
On the **Export** dialog

Click the **Java** folder and select **Runnable Jar file**, and **Next**



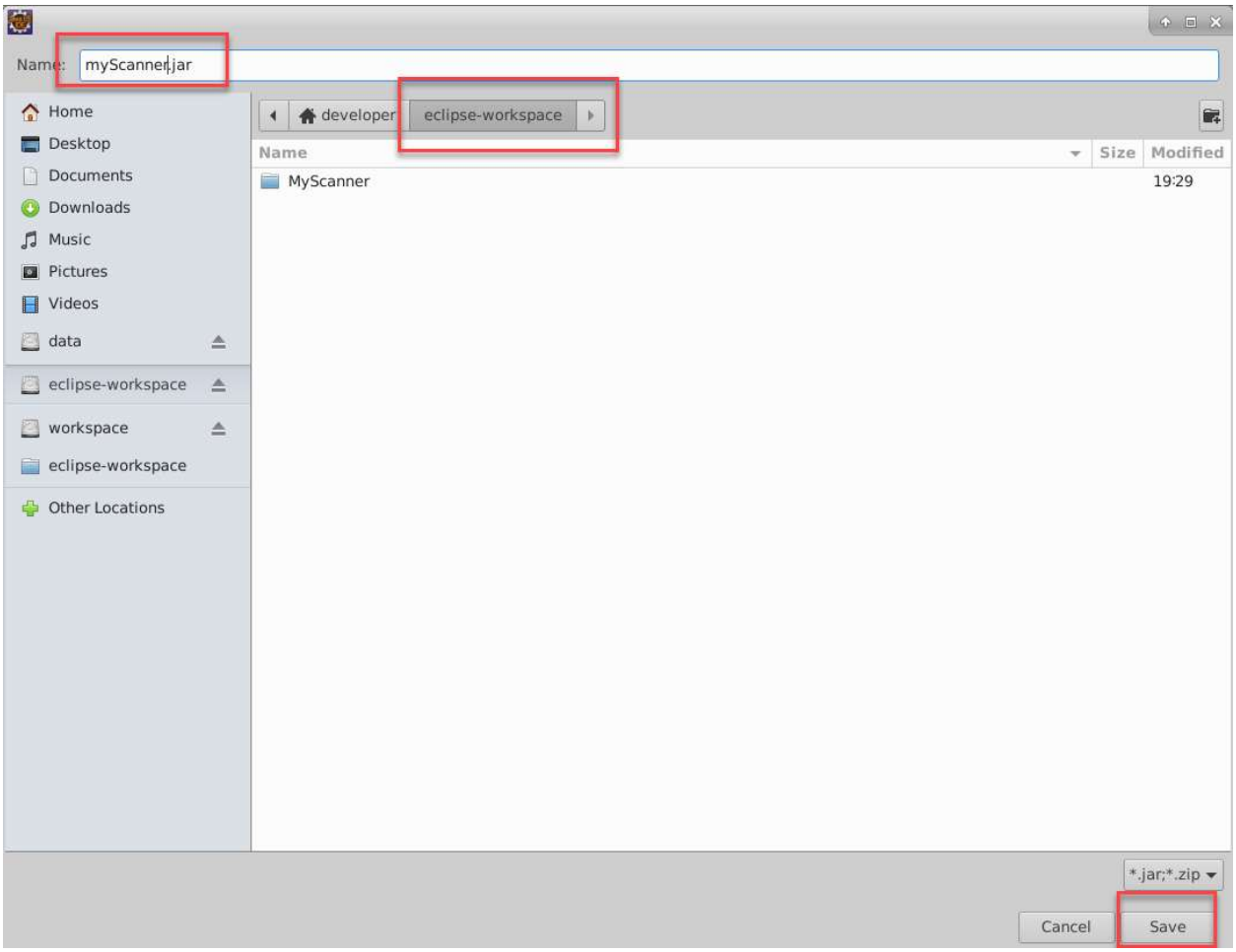
From **Launch configuration** dropdown, select: ScannerConsole – MyScanner, or the available Scanner Launch Configuration

Click **Browse** next to the **Export destination** field, to select the directory for the jar file.

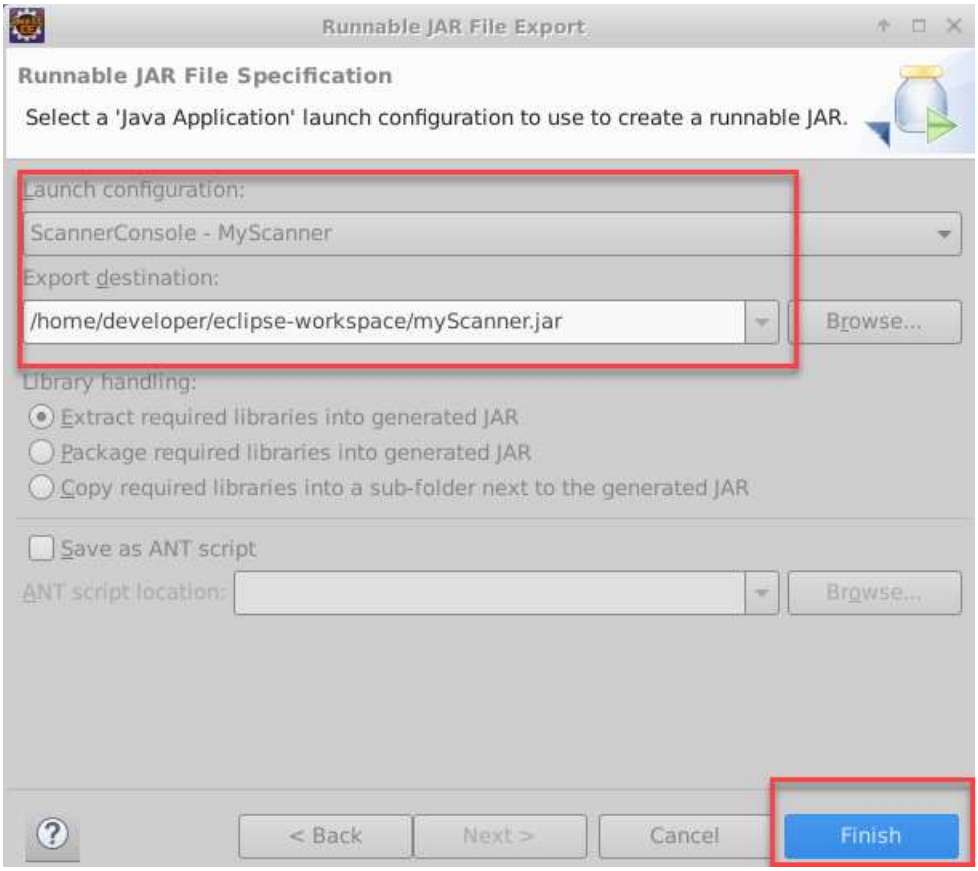


Name the jar file: myScanner.jar

Click **eclipse-workspace** and **Save**.

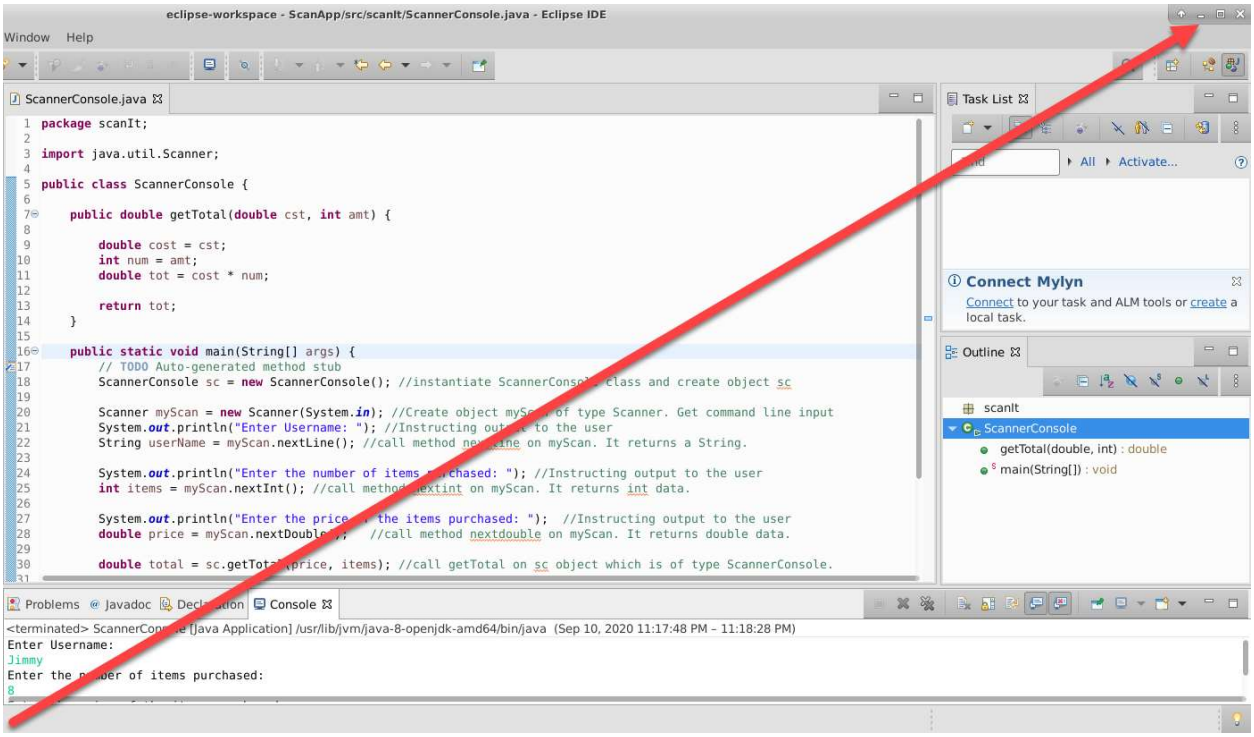


Make note of the **Export destination** directory and click **Finish**

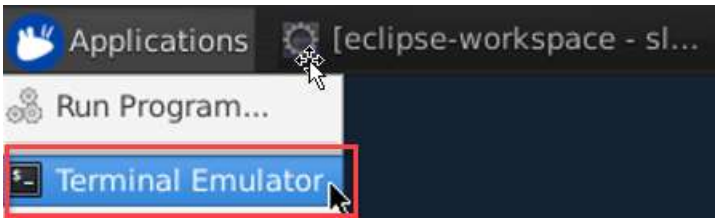


Run the jar from the command line.

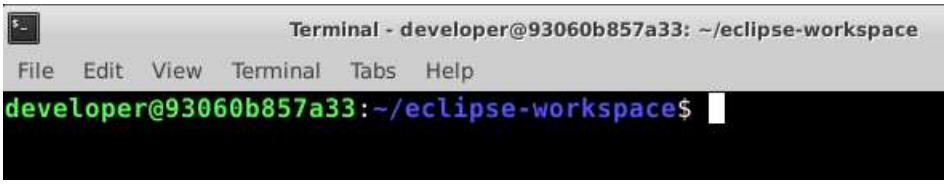
Minimize Eclipse. Click the minimize button in the upper right hand corner. This brings you back to the Desktop.



On the Desktop, click the Applications button, and **Terminal Emulator**.

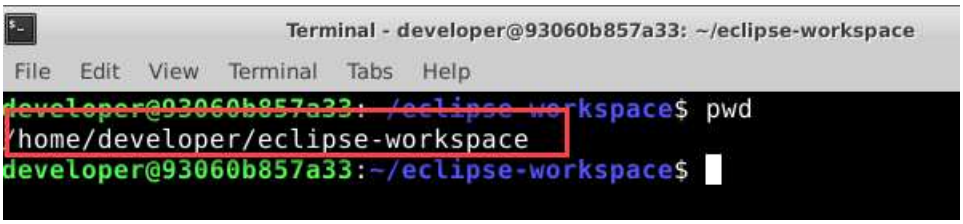


A terminal will open.



In the terminal, run command: **pwd**

Make sure the present working directory (**pwd**) is: **/home/developer/eclipse-workspace**



To run the jar file enter the command: java -jar ./myScanner.jar

```
Terminal - developer@dbbf138c2cb5: ~/eclipse-workspace
File  Edit  View  Terminal  Tabs  Help
developer@dbbf138c2cb5:~/eclipse-workspace$ java -jar ./myScanner.jar
```

Enter a name, a number of items, and a price. Observe the result

```
Terminal - developer@dbbf138c2cb5: ~/eclipse-wo
File  Edit  View  Terminal  Tabs  Help
developer@dbbf138c2cb5:~/eclipse-workspace$ java -jar ./myScanner.jar
Enter Username:
Jimmy
Enter the number of items purchased:
14
Enter the price of the items purchased:
8.23
Greetings: Jimmy
The number of items is: 14
The price is: 8.23
The total is: 115.22
developer@dbbf138c2cb5:~/eclipse-workspace$
```

End of Lab