Java – Fundamentals Core Platform

Introduction

Java
Has two identities, first is know to be a programming language. Second known to be a runtime environment .

Java FX: is a software platform for creating and delivering desktop applications, as well as rich internet applications (RIAs) that can run across a wide variety of devices

Java EE (Enterprise Edition): used to create highly scalable applications referred as enterprise class applications.

Java ME: Subset of Java runtime environment intended for embedded systems. This is widely used with 'Internet of things'

Internet of things: refers to autonomous (independent) or semiautonomous (separate) systems that are connected via the internet.

Streams

Stream is an ordered sequence of data

Streams are unidirectional. They can 'read from', or 'write to'. No single stream does both

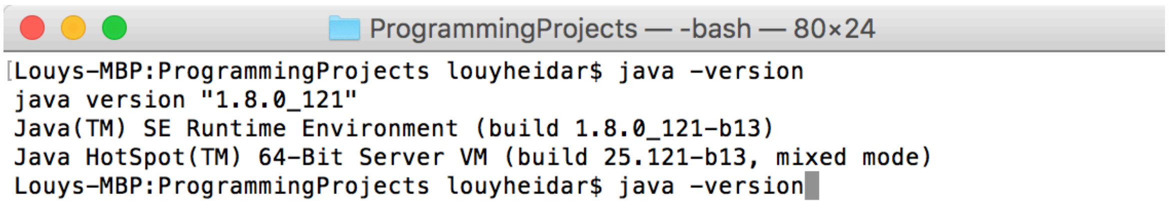
There are two categories of streams, byte steams or text streams

Java Fundamentals – The Java Language

Java is not compare to low level programming languages like C

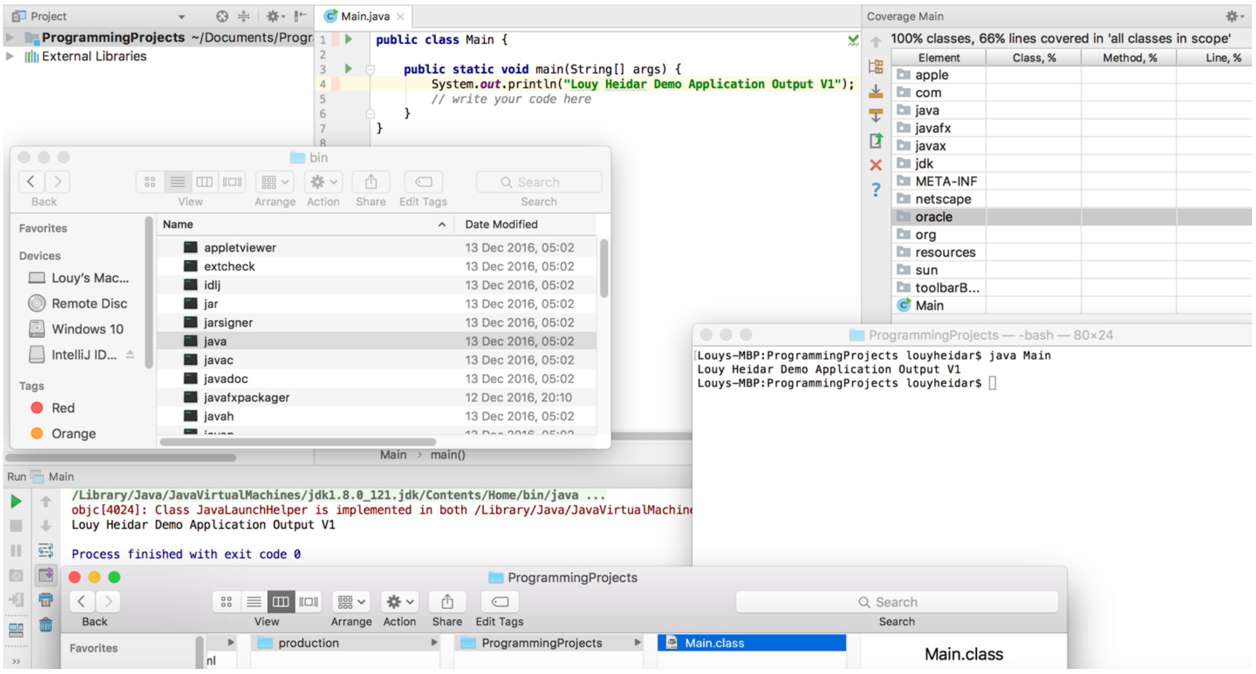
When a C program is compiled it produces an application that can run directly on the host computer. Java uses an abstraction called Blake Codes that is platform independent i.e. to not be limited to a particular host environment.

Java Runtime Environment provides what is needed to Execute Java Apps.



```
[Louys-MBP:ProgrammingProjects louyheidar$ java -version
java version "1.8.0_121"
Java(TM) SE Runtime Environment (build 1.8.0_121-b13)
Java HotSpot(TM) 64-Bit Server VM (build 25.121-b13, mixed mode)
Louys-MBP:ProgrammingProjects louyheidar$ java -version
```

Java – Running Demo Application Manually via Terminal/Command Prompt



```java
public class Main {

    public static void main(String[] args) {
        System.out.println("Louy Heidar Demo Application Output V1");
        // write your code here
    }
}
```

Coverage Main — 100% classes, 66% lines covered in 'all classes in scope'

Element	Class, %	Method, %	Line, %
apple			
com			
java			
javafx			
javax			
jdk			
META-INF			
netscape			
oracle			
org			
resources			
sun			
toolbarB...			
Main			

bin

Name	Date Modified
appletviewer	13 Dec 2016, 05:02
extcheck	13 Dec 2016, 05:02
idlj	13 Dec 2016, 05:02
jar	13 Dec 2016, 05:02
jarsigner	13 Dec 2016, 05:02
java	13 Dec 2016, 05:02
javac	13 Dec 2016, 05:02
javadoc	13 Dec 2016, 05:02
javafxpackager	12 Dec 2016, 20:10
javah	13 Dec 2016, 05:02

ProgrammingProjects — -bash — 80×24

```
Louys-MBP:ProgrammingProjects louyheidar$ java Main
Louy Heidar Demo Application Output V1
Louys-MBP:ProgrammingProjects louyheidar$
```

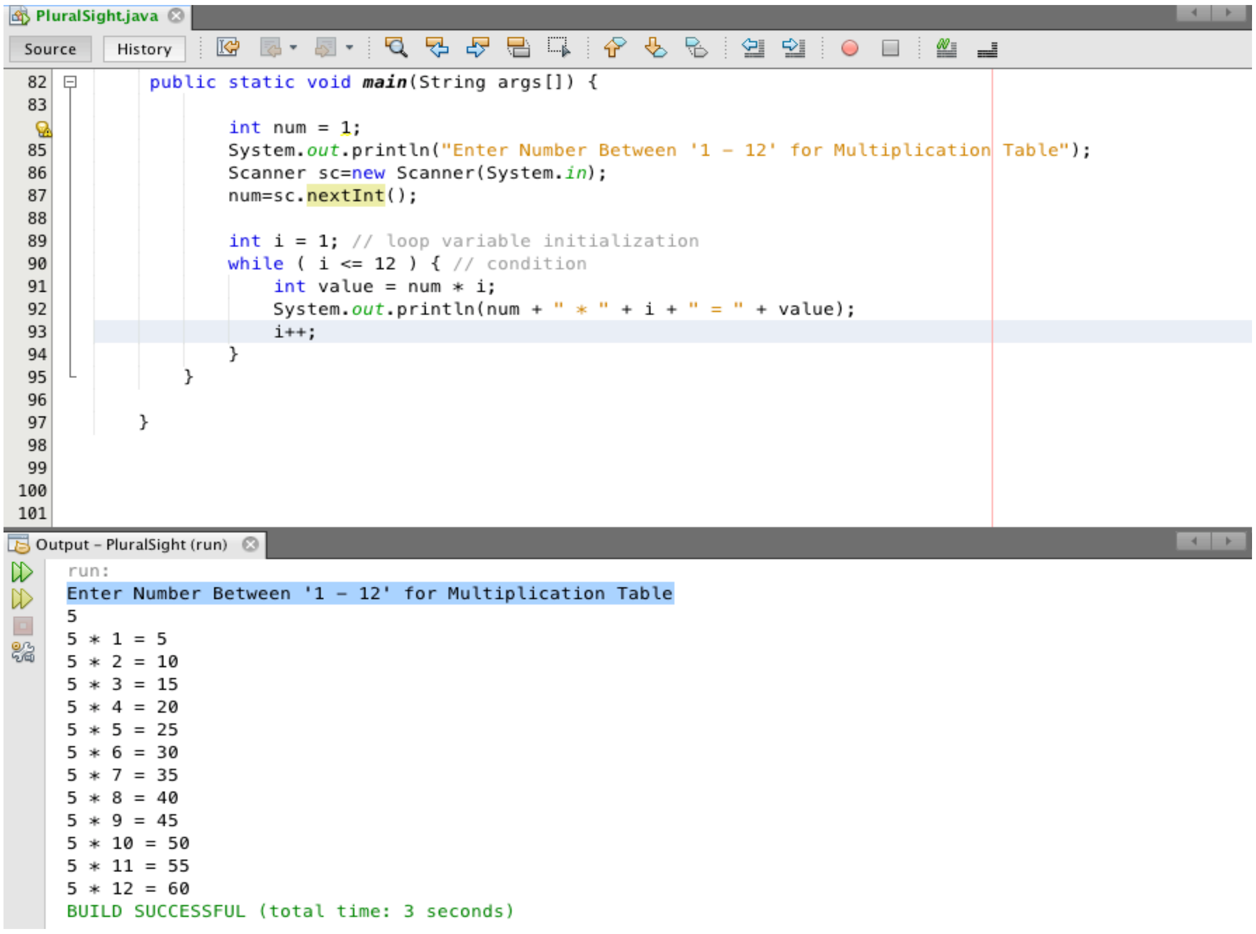
Run Main

```
/Library/Java/JavaVirtualMachines/jdk1.8.0_121.jdk/Contents/Home/bin/java ...
objc[4024]: Class JavaLaunchHelper is implemented in both /Library/Java/JavaVirtualMachine
Louy Heidar Demo Application Output V1

Process finished with exit code 0
```

ProgrammingProjects: production ▸ ProgrammingProjects ▸ Main.class

Java – Looping

Multiplication Table (Accepting User Input)

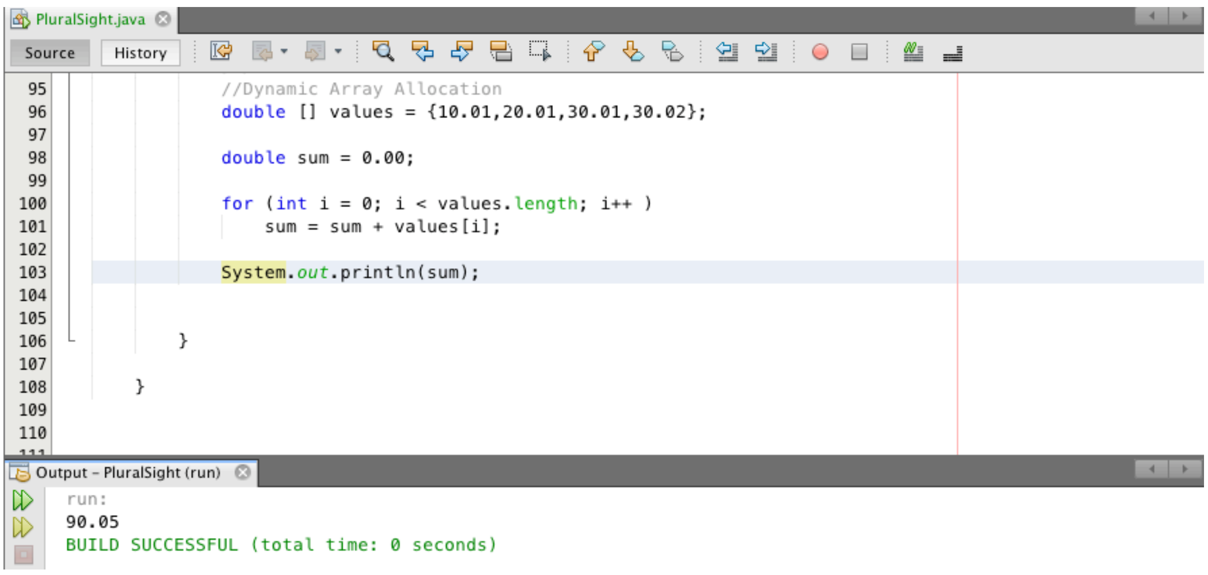


```java
    public static void main(String args[]) {

        int num = 1;
        System.out.println("Enter Number Between '1 - 12' for Multiplication Table");
        Scanner sc=new Scanner(System.in);
        num=sc.nextInt();

        int i = 1; // loop variable initialization
        while ( i <= 12 ) { // condition
            int value = num * i;
            System.out.println(num + " * " + i + " = " + value);
            i++;
        }
    }

}
```

Output - PluralSight (run)

```
run:
Enter Number Between '1 - 12' for Multiplication Table
5
5 * 1 = 5
5 * 2 = 10
5 * 3 = 15
5 * 4 = 20
5 * 5 = 25
5 * 6 = 30
5 * 7 = 35
5 * 8 = 40
5 * 9 = 45
5 * 10 = 50
5 * 11 = 55
5 * 12 = 60
BUILD SUCCESSFUL (total time: 3 seconds)
```

Java – Arrays



PluralSight.java

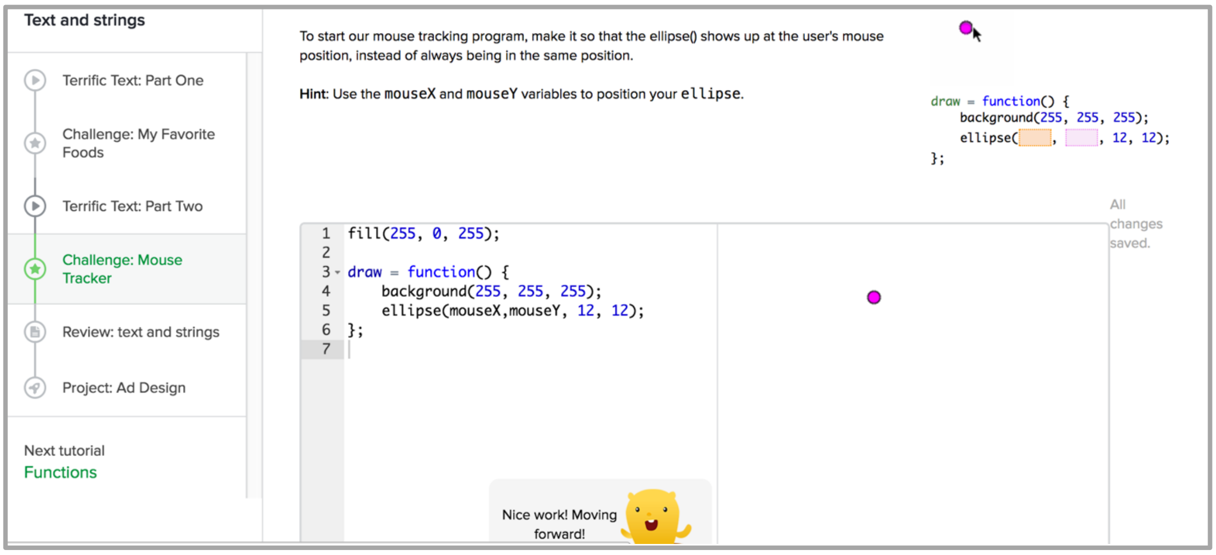
Source | History

```
            //Dynamic Array Allocation
            double [] values = {10.01,20.01,30.01,30.02};

            double sum = 0.00;

            for (int i = 0; i < values.length; i++ )
                sum = sum + values[i];

            System.out.println(sum);


        }

    }
```

Output – PluralSight (run)

```
run:
90.05
BUILD SUCCESSFUL (total time: 0 seconds)
```

JavaScript – Mouse Tracker



Text and strings

- Terrific Text: Part One
- Challenge: My Favorite Foods
- Terrific Text: Part Two
- Challenge: Mouse Tracker
- Review: text and strings
- Project: Ad Design

Next tutorial
Functions

To start our mouse tracking program, make it so that the ellipse() shows up at the user's mouse position, instead of always being in the same position.

Hint: Use the `mouseX` and `mouseY` variables to position your `ellipse`.

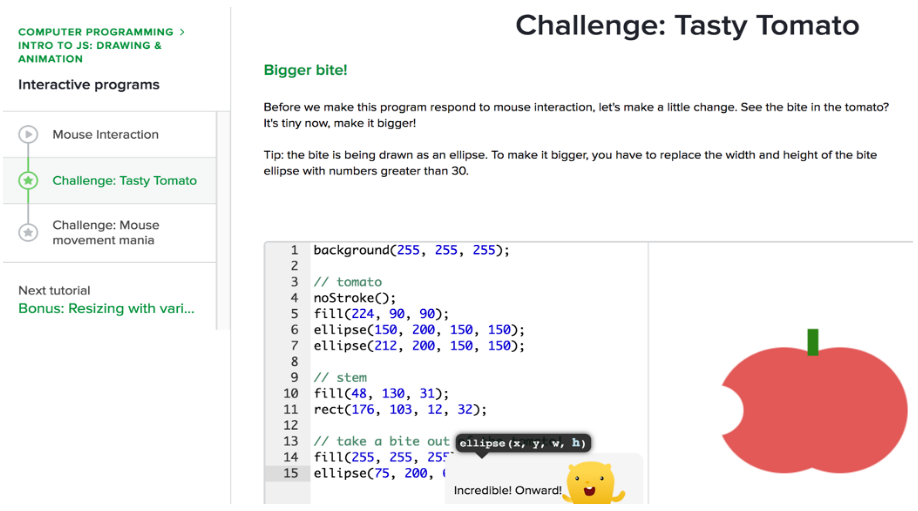
```
draw = function() {
    background(255, 255, 255);
    ellipse(   ,    , 12, 12);
};
```

All changes saved.

```
fill(255, 0, 255);

draw = function() {
    background(255, 255, 255);
    ellipse(mouseX,mouseY, 12, 12);
};

```

Nice work! Moving forward!

JavaScript – Interactive Applications



COMPUTER PROGRAMMING >
INTRO TO JS: DRAWING & ANIMATION

Interactive programs

- Mouse Interaction
- Challenge: Tasty Tomato
- Challenge: Mouse movement mania

Next tutorial
Bonus: Resizing with vari...

Challenge: Tasty Tomato

Bigger bite!

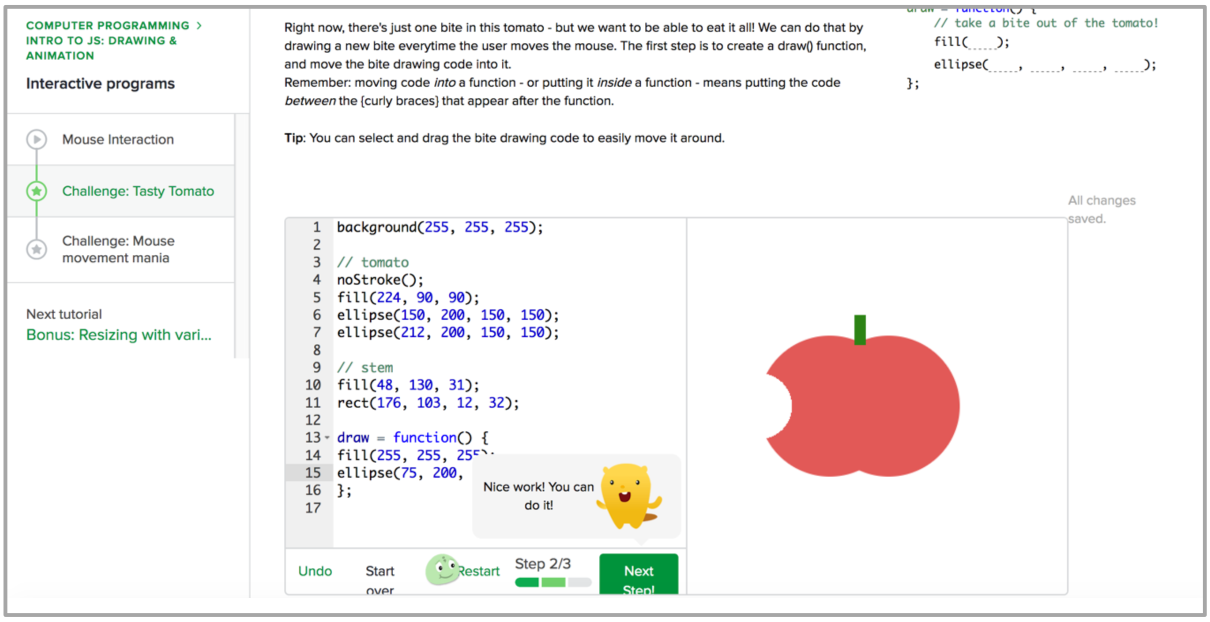
Before we make this program respond to mouse interaction, let's make a little change. See the bite in the tomato? It's tiny now, make it bigger!

Tip: the bite is being drawn as an ellipse. To make it bigger, you have to replace the width and height of the bite ellipse with numbers greater than 30.

```
background(255, 255, 255);

// tomato
noStroke();
fill(224, 90, 90);
ellipse(150, 200, 150, 150);
ellipse(212, 200, 150, 150);

// stem
fill(48, 130, 31);
rect(176, 103, 12, 32);

// take a bite out
fill(255, 255, 255
ellipse(75, 200, (
```

ellipse(x, y, w, h)

Incredible! Onward!



COMPUTER PROGRAMMING > INTRO TO JS: DRAWING & ANIMATION

Interactive programs

Mouse Interaction

Challenge: Tasty Tomato

Challenge: Mouse movement mania

Next tutorial
Bonus: Resizing with vari...

Right now, there's just one bite in this tomato - but we want to be able to eat it all! We can do that by drawing a new bite everytime the user moves the mouse. The first step is to create a draw() function, and move the bite drawing code into it.
Remember: moving code *into* a function - or putting it *inside* a function - means putting the code *between* the {curly braces} that appear after the function.

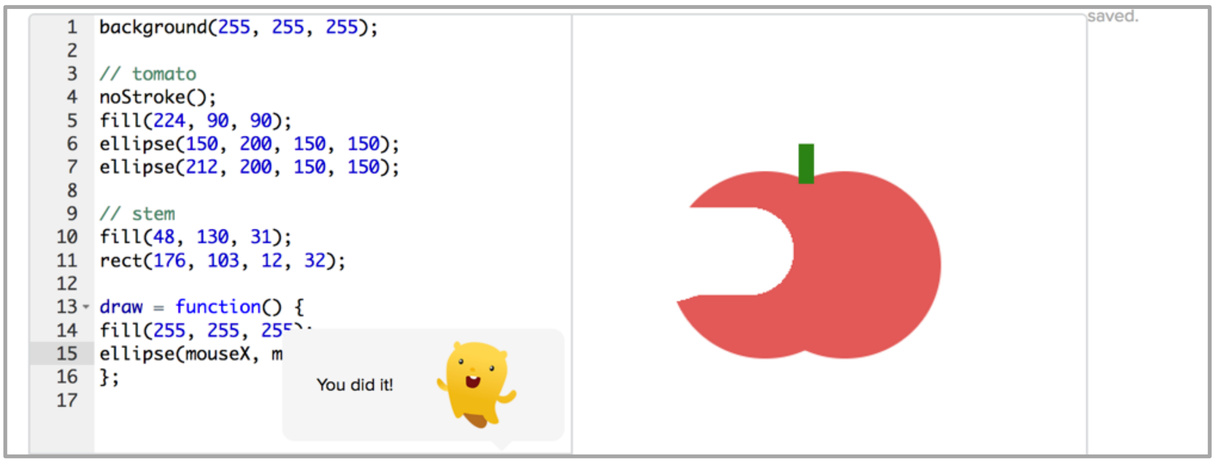
Tip: You can select and drag the bite drawing code to easily move it around.

```
   // take a bite out of the tomato!
   fill(____);
   ellipse(____, ____, ____, ____);
};
```

All changes saved.

```
background(255, 255, 255);

// tomato
noStroke();
fill(224, 90, 90);
ellipse(150, 200, 150, 150);
ellipse(212, 200, 150, 150);

// stem
fill(48, 130, 31);
rect(176, 103, 12, 32);

draw = function() {
fill(255, 255, 25
ellipse(75, 200,
};
```

JavaScript – Responsive JS Mouse Events

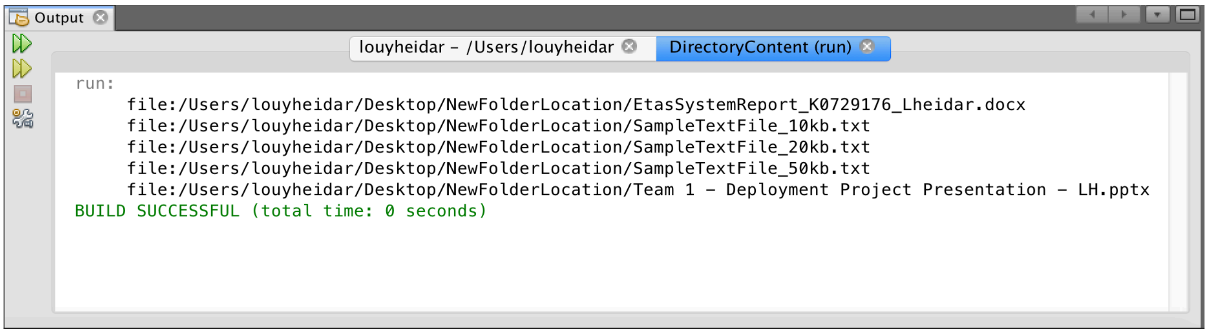


```
background(255, 255, 255);

// tomato
noStroke();
fill(224, 90, 90);
ellipse(150, 200, 150, 150);
ellipse(212, 200, 150, 150);

// stem
fill(48, 130, 31);
rect(176, 103, 12, 32);

draw = function() {
fill(255, 255, 25
ellipse(mouseX, m
};
```

Java – Listing Directory Content

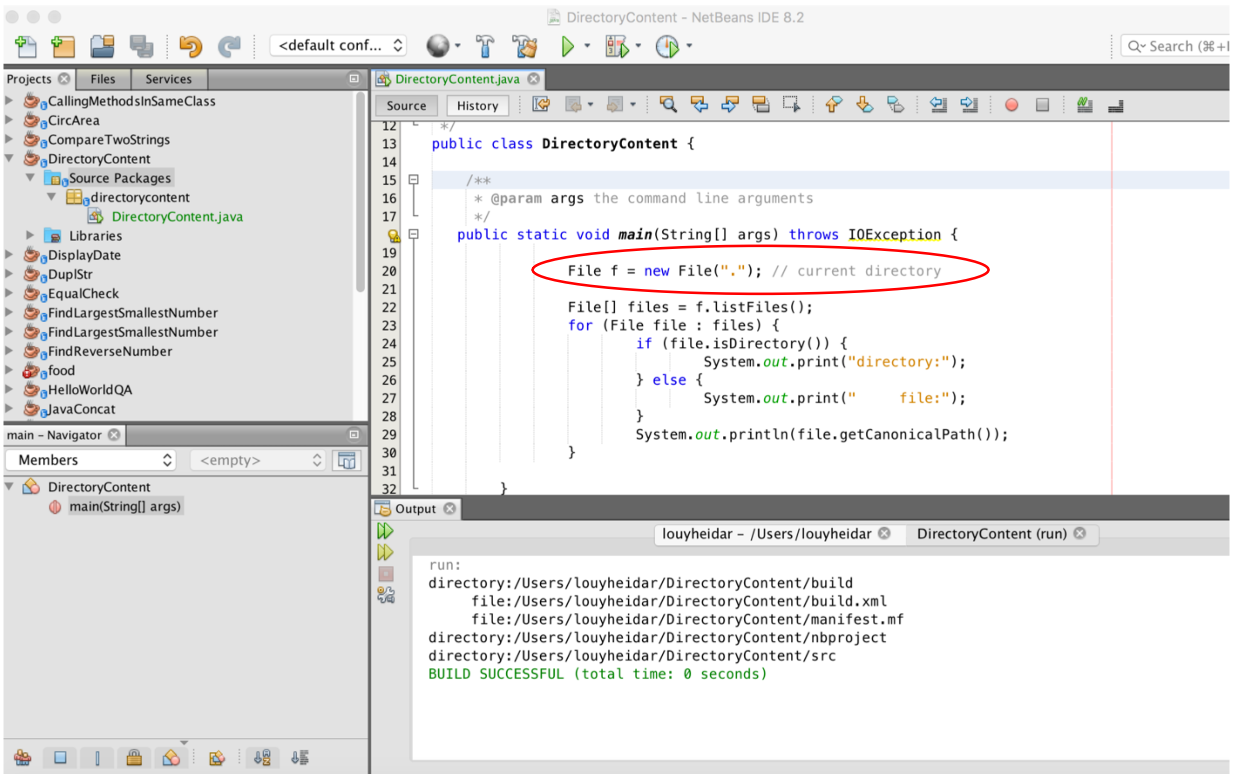
isDirectory() method used to determine if the object is a directory. If it is a directory, an array of the File objects within it can be obtained by a call to listFiles().

DirectoryContents class gets a File object to the current directory. It gets an array of all the File objects within the current directory by calling f.listFiles(). It displays whether each File object is a file or a directory and displays its path.



```
run:
        file:/Users/louyheidar/Desktop/NewFolderLocation/EtasSystemReport_K0729176_Lheidar.docx
        file:/Users/louyheidar/Desktop/NewFolderLocation/SampleTextFile_10kb.txt
        file:/Users/louyheidar/Desktop/NewFolderLocation/SampleTextFile_20kb.txt
        file:/Users/louyheidar/Desktop/NewFolderLocation/SampleTextFile_50kb.txt
        file:/Users/louyheidar/Desktop/NewFolderLocation/Team 1 – Deployment Project Presentation – LH.pptx
BUILD SUCCESSFUL (total time: 0 seconds)
```

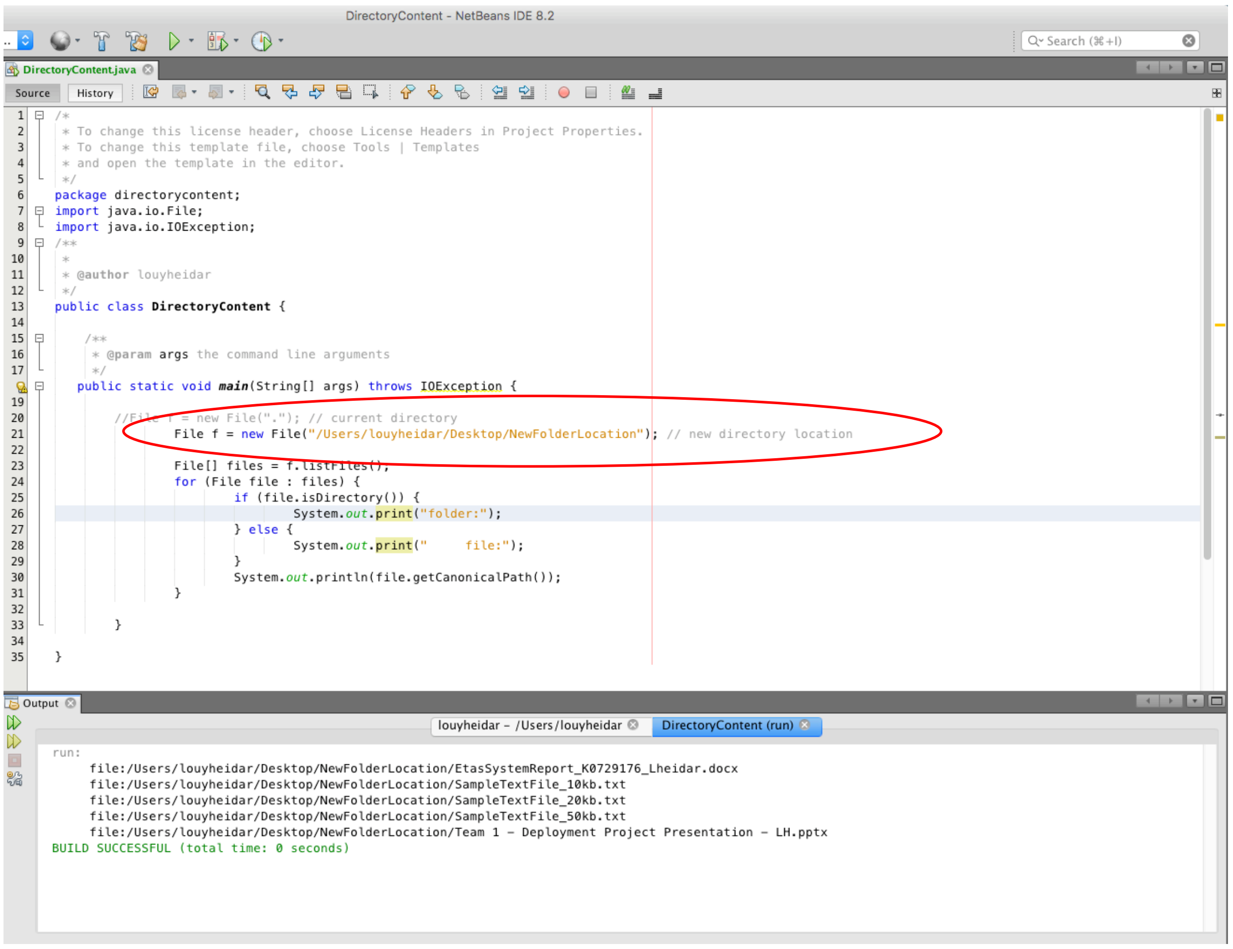
Java – Locating & Listing Files and Folders - Current Directory



```java
  */
public class DirectoryContent {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) throws IOException {

            File f = new File("."); // current directory

            File[] files = f.listFiles();
            for (File file : files) {
                if (file.isDirectory()) {
                    System.out.print("directory:");
                } else {
                    System.out.print("     file:");
                }
                System.out.println(file.getCanonicalPath());
            }

        }
```

```
run:
directory:/Users/louyheidar/DirectoryContent/build
     file:/Users/louyheidar/DirectoryContent/build.xml
     file:/Users/louyheidar/DirectoryContent/manifest.mf
directory:/Users/louyheidar/DirectoryContent/nbproject
directory:/Users/louyheidar/DirectoryContent/src
BUILD SUCCESSFUL (total time: 0 seconds)
```

Java – Locating & Listing Files and Folders - Specified Directory



```java
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package directorycontent;
import java.io.File;
import java.io.IOException;
/**
 *
 * @author louyheidar
 */
public class DirectoryContent {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) throws IOException {

        //File f = new File("."); // current directory
                File f = new File("/Users/louyheidar/Desktop/NewFolderLocation"); // new directory location

                File[] files = f.listFiles();
                for (File file : files) {
                        if (file.isDirectory()) {
                                System.out.print("folder:");
                        } else {
                                System.out.print("     file:");
                        }
                        System.out.println(file.getCanonicalPath());
                }

        }

}
```

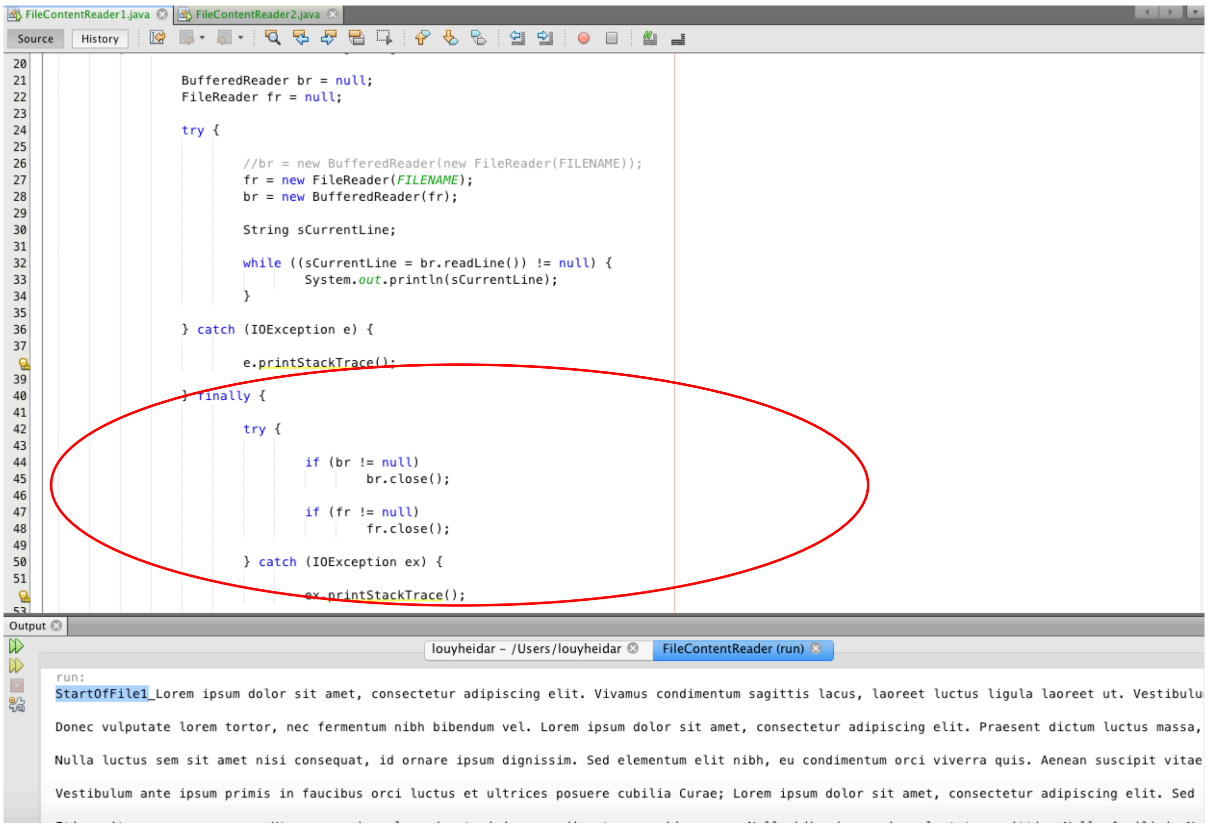
```
run:
     file:/Users/louyheidar/Desktop/NewFolderLocation/EtasSystemReport_K0729176_Lheidar.docx
     file:/Users/louyheidar/Desktop/NewFolderLocation/SampleTextFile_10kb.txt
     file:/Users/louyheidar/Desktop/NewFolderLocation/SampleTextFile_20kb.txt
     file:/Users/louyheidar/Desktop/NewFolderLocation/SampleTextFile_50kb.txt
     file:/Users/louyheidar/Desktop/NewFolderLocation/Team 1 – Deployment Project Presentation – LH.pptx
BUILD SUCCESSFUL (total time: 0 seconds)
```

Java – Reading Files (Buffered Reader)

"Try" and "catch" - represent handling of exceptions due to data or coding errors during the program execution. 'Try' block is the block of code in which exceptions occur. 'Catch' block catches and handles try block exceptions.

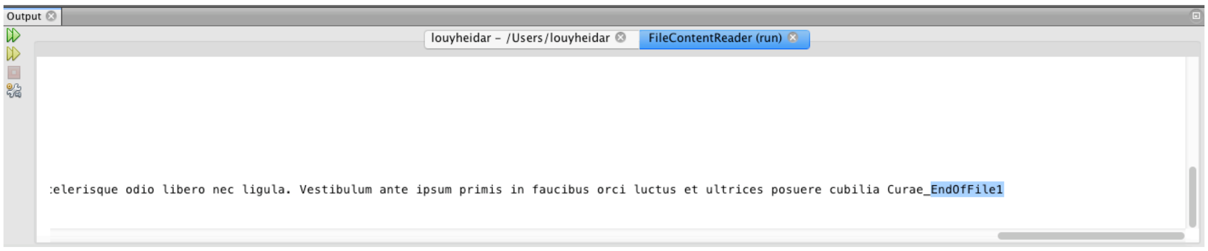
'FileContentReader1.java' (Example 1): Manually closes the file as a way of handling the file itself, and the content within that file.

Example 1



```java
BufferedReader br = null;
FileReader fr = null;

try {

        //br = new BufferedReader(new FileReader(FILENAME));
        fr = new FileReader(FILENAME);
        br = new BufferedReader(fr);

        String sCurrentLine;

        while ((sCurrentLine = br.readLine()) != null) {
                System.out.println(sCurrentLine);
        }

} catch (IOException e) {

        e.printStackTrace();

} finally {

        try {

                if (br != null)
                        br.close();

                if (fr != null)
                        fr.close();

        } catch (IOException ex) {

                ex.printStackTrace();
```

```
run:
StartOfFile1_Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus condimentum sagittis lacus, laoreet luctus ligula laoreet ut. Vestibulu

Donec vulputate lorem tortor, nec fermentum nibh bibendum vel. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent dictum luctus massa,

Nulla luctus sem sit amet nisi consequat, id ornare ipsum dignissim. Sed elementum elit nibh, eu condimentum orci viverra quis. Aenean suscipit vitae

Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed
```

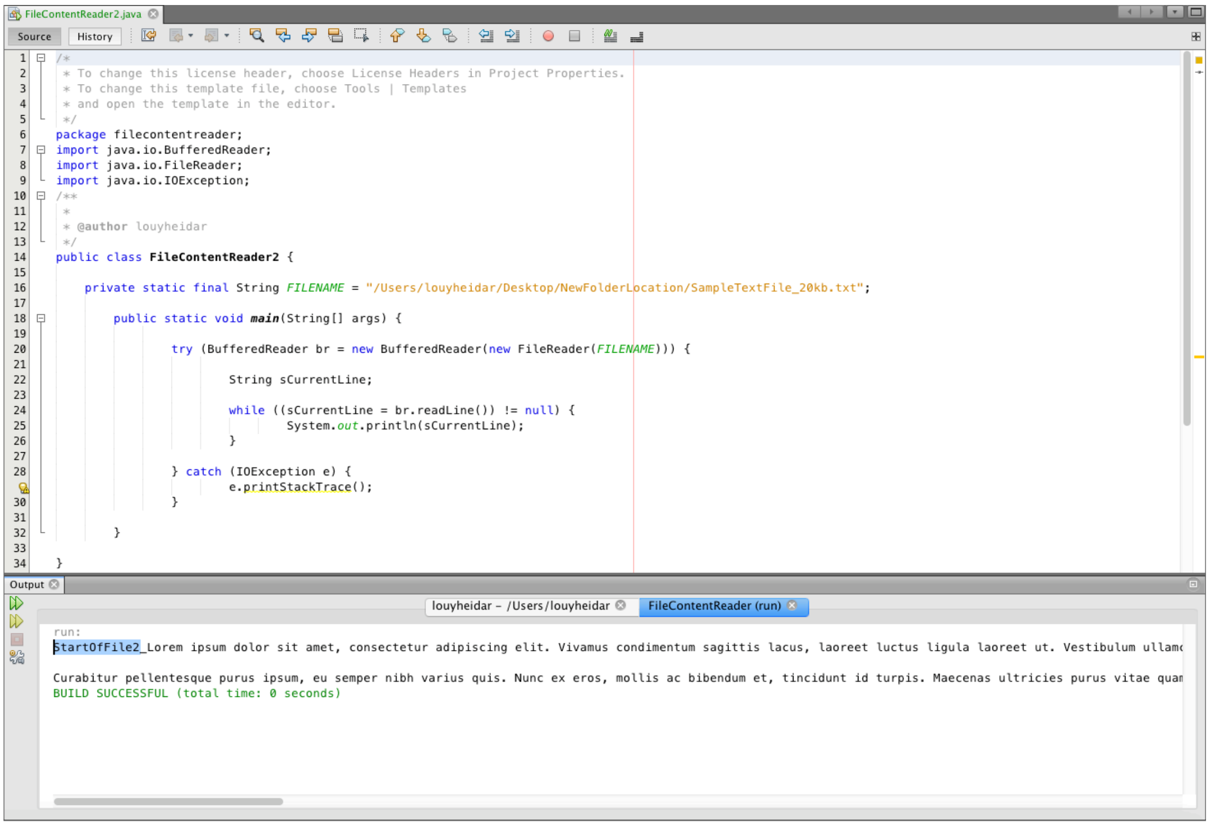


```
:elerisque odio libero nec ligula. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae_EndOfFile1
```

'FileContentReader2.java' auto closes the file which is a more efficient and up to date concept for programming. It reduces the room for user error i.e. programmers forgetting to manually close the file.

Example 2 below demonstrates auto close for file reader

Example 2

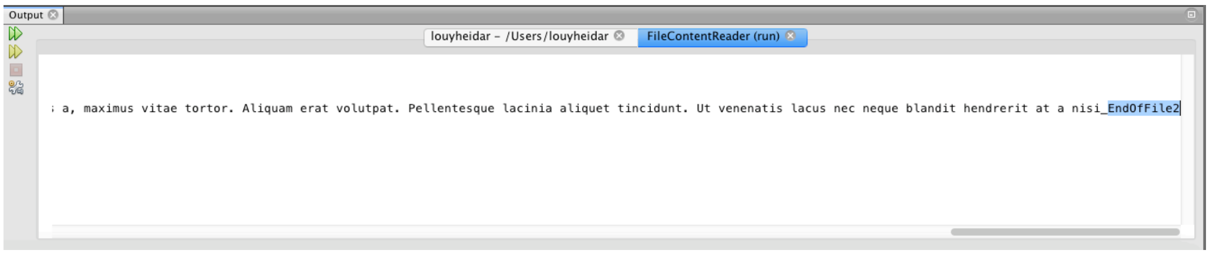


FileContentReader2.java

```java
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package filecontentreader;
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;
/**
 *
 * @author louyheidar
 */
public class FileContentReader2 {

    private static final String FILENAME = "/Users/louyheidar/Desktop/NewFolderLocation/SampleTextFile_20kb.txt";

        public static void main(String[] args) {

                try (BufferedReader br = new BufferedReader(new FileReader(FILENAME))) {

                        String sCurrentLine;

                        while ((sCurrentLine = br.readLine()) != null) {
                                System.out.println(sCurrentLine);
                        }

                } catch (IOException e) {
                        e.printStackTrace();
                }

        }

}
```

Output — louyheidar – /Users/louyheidar | FileContentReader (run)

```
run:
StartOfFile2_Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus condimentum sagittis lacus, laoreet luctus ligula laoreet ut. Vestibulum ullamc

Curabitur pellentesque purus ipsum, eu semper nibh varius quis. Nunc ex eros, mollis ac bibendum et, tincidunt id turpis. Maecenas ultricies purus vitae quan
BUILD SUCCESSFUL (total time: 0 seconds)
```



Output — louyheidar – /Users/louyheidar | FileContentReader (run)

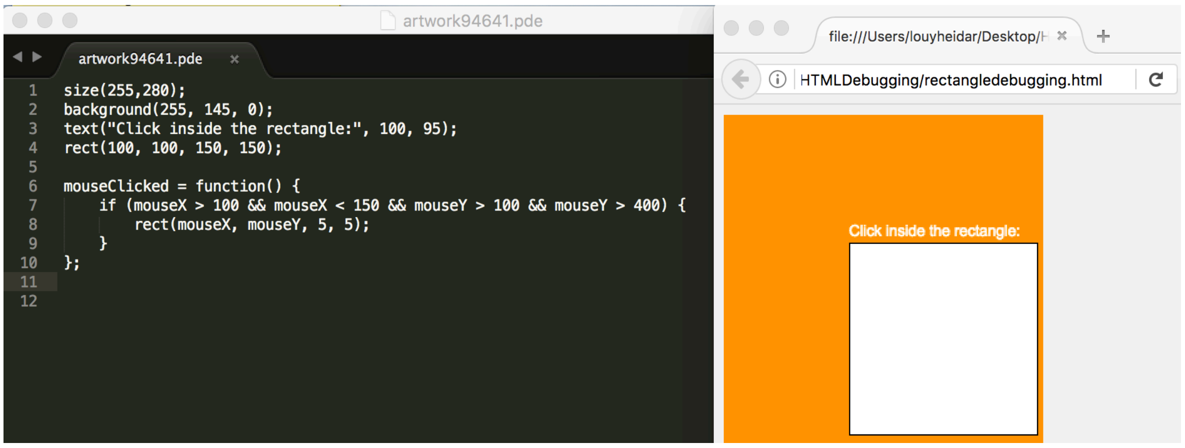
```
a, maximus vitae tortor. Aliquam erat volutpat. Pellentesque lacinia aliquet tincidunt. Ut venenatis lacus nec neque blandit hendrerit at a nisi_EndOfFile2
```

JavaScript – Debugging with Println Statements

Debugging: identify and remove errors from applications

Problem: White Rectangle object not responding to user input i.e. shapes not appearing on mouse click

Example Code (Not Working)



artwork94641.pde

```javascript
size(255,280);
background(255, 145, 0);
text("Click inside the rectangle:", 100, 95);
rect(100, 100, 150, 150);

mouseClicked = function() {
    if (mouseX > 100 && mouseX < 150 && mouseY > 100 && mouseY > 400) {
        rect(mouseX, mouseY, 5, 5);
    }
};
```

file:///Users/louyheidar/Desktop/HTMLDebugging/rectangledebugging.html
Click inside the rectangle:

Debugging Steps

<u>Step 1</u>

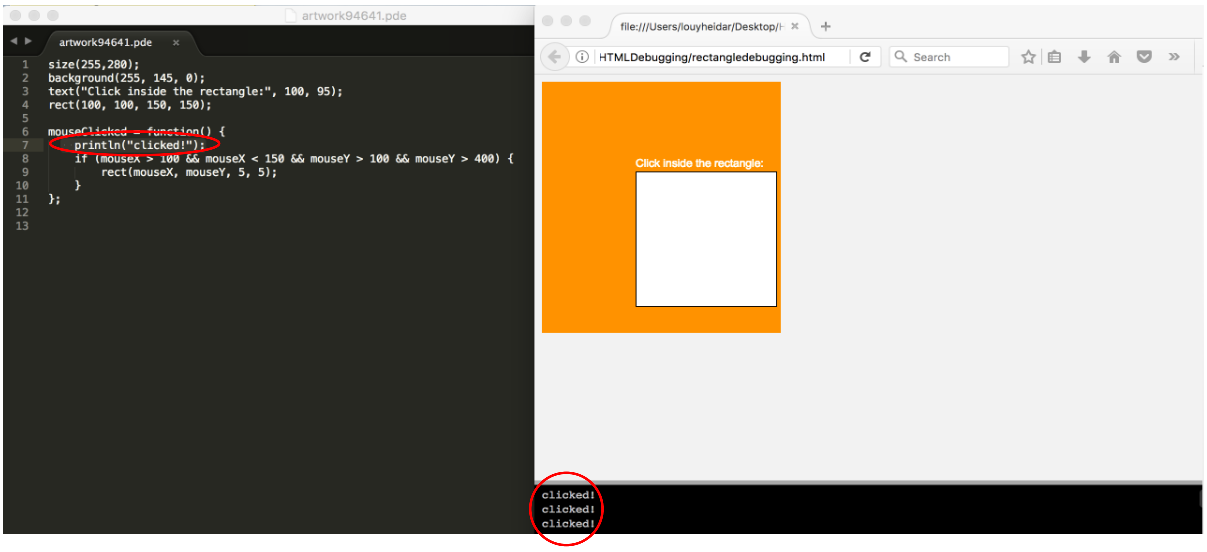
Identify and check to see whether the ‘mouseClicked = function()’ is being called i.e. this function should be called with every mouse click

<u>Step 2</u>

Inserting println method underneath ‘mouseClicked’ function will verify whether that function is working.

The word ‘clicked’ should appear within console output in browser with every mouse click.

Output message proves that the ‘mouseClicked’ function is being called and working.


artwork94641.pde
file:///Users/louyheidar/Desktop/H
HTMLDebugging/rectangledebugging.html
Search
Click inside the rectangle:

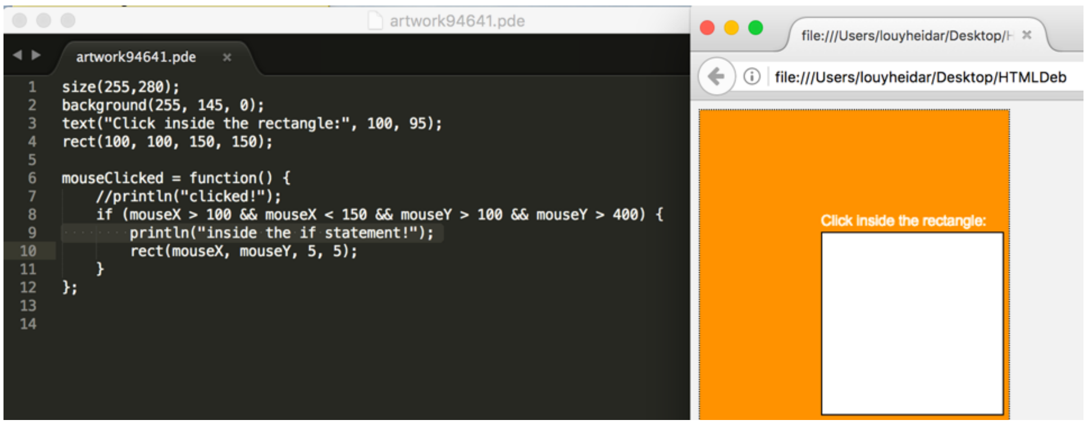
```
size(255,280);
background(255, 145, 0);
text("Click inside the rectangle:", 100, 95);
rect(100, 100, 150, 150);

mouseClicked = function() {
    println("clicked!");
    if (mouseX > 100 && mouseX < 150 && mouseY > 100 && mouseY > 400) {
        rect(mouseX, mouseY, 5, 5);
    }
};
```

```
clicked!
clicked!
clicked!
```

<u>Step 3</u>

Inserting a println statement inside the if statement to verify whether there is bug or problem with what we are displaying.

No message is printed within browser console on mouse click, which verifies that the if statement is never true. This narrows down the cause i.e. problem lies within the if condition


artwork94641.pde
file:///Users/louyheidar/Desktop/H
file:///Users/louyheidar/Desktop/HTMLDeb
Click inside the rectangle:

```
size(255,280);
background(255, 145, 0);
text("Click inside the rectangle:", 100, 95);
rect(100, 100, 150, 150);

mouseClicked = function() {
    //println("clicked!");
    if (mouseX > 100 && mouseX < 150 && mouseY > 100 && mouseY > 400) {
        println("inside the if statement!");
        rect(mouseX, mouseY, 5, 5);
    }
};
```

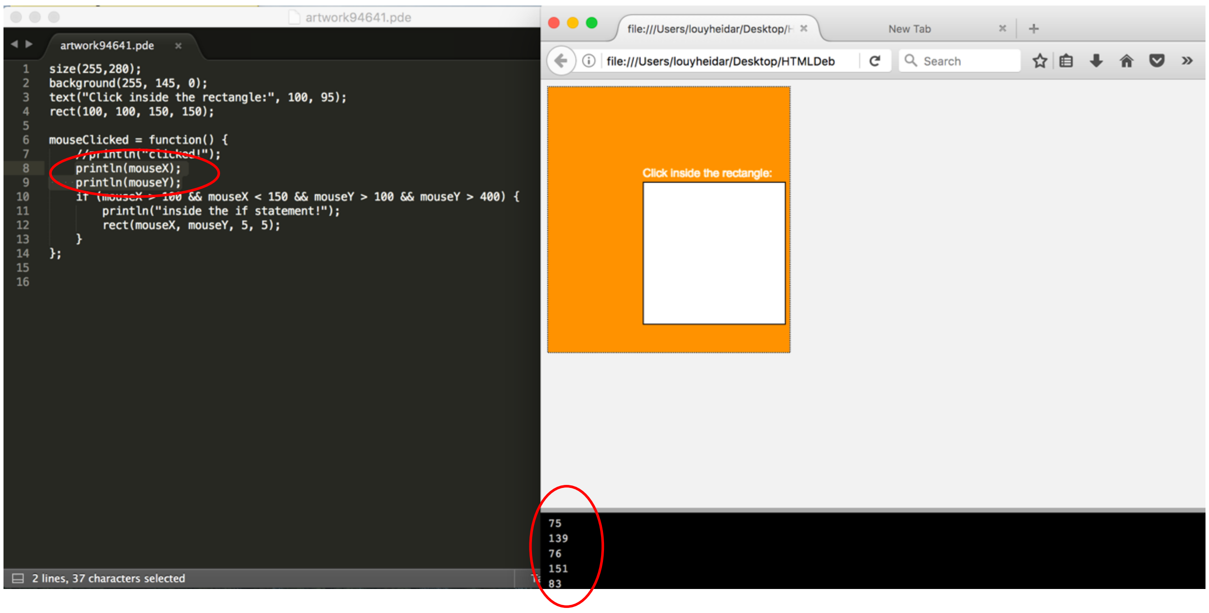
Step 4

Determine what is wrong with the if condition i.e. why is the condition never true.

Print the values that are being checked, and check that value output within console window looks reasonable

Lines Added:
println(mouseX);
println(mouseY);

Console output functions and display correctly. This verifies that there are no problems with the checked values.



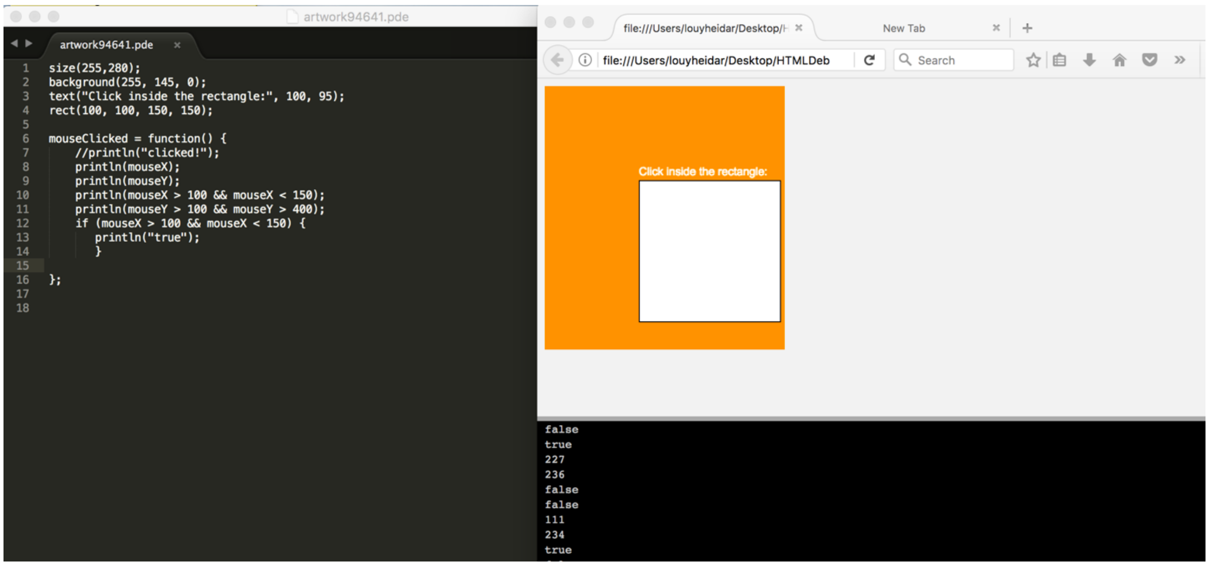
```
size(255,280);
background(255, 145, 0);
text("Click inside the rectangle:", 100, 95);
rect(100, 100, 150, 150);

mouseClicked = function() {
    //println("clicked!");
    println(mouseX);
    println(mouseY);
    if (mouseX > 100 && mouseX < 150 && mouseY > 100 && mouseY > 400) {
        println("inside the if statement!");
        rect(mouseX, mouseY, 5, 5);
    }
};
```

```
75
139
76
151
83
```

artwork94641.pde
2 lines, 37 characters selected
file:///Users/louyheidar/Desktop/HTMLDeb
New Tab
Search
Click inside the rectangle:

Step 5

Split 'if condition' to debug separately i.e. debug if conditions as two separate sections



```
size(255,280);
background(255, 145, 0);
text("Click inside the rectangle:", 100, 95);
rect(100, 100, 150, 150);

mouseClicked = function() {
    //println("clicked!");
    println(mouseX);
    println(mouseY);
    println(mouseX > 100 && mouseX < 150);
    println(mouseY > 100 && mouseY > 400);
    if (mouseX > 100 && mouseX < 150) {
       println("true");
       }

};
```

```
false
true
227
236
false
false
111
234
true
```

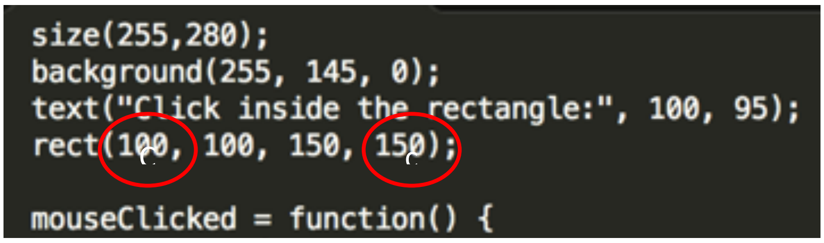
artwork94641.pde
file:///Users/louyheidar/Desktop/HTMLDeb
New Tab
Search
Click inside the rectangle:

Step 6

Match values of rectangle with the canvas coordinates to determine where the position of the rectangle is located and where the position of the canvas is.

Correct reading:
mouseY > top of rectangle && mouseY< bottom of rectangle

Bottom of rectangle:

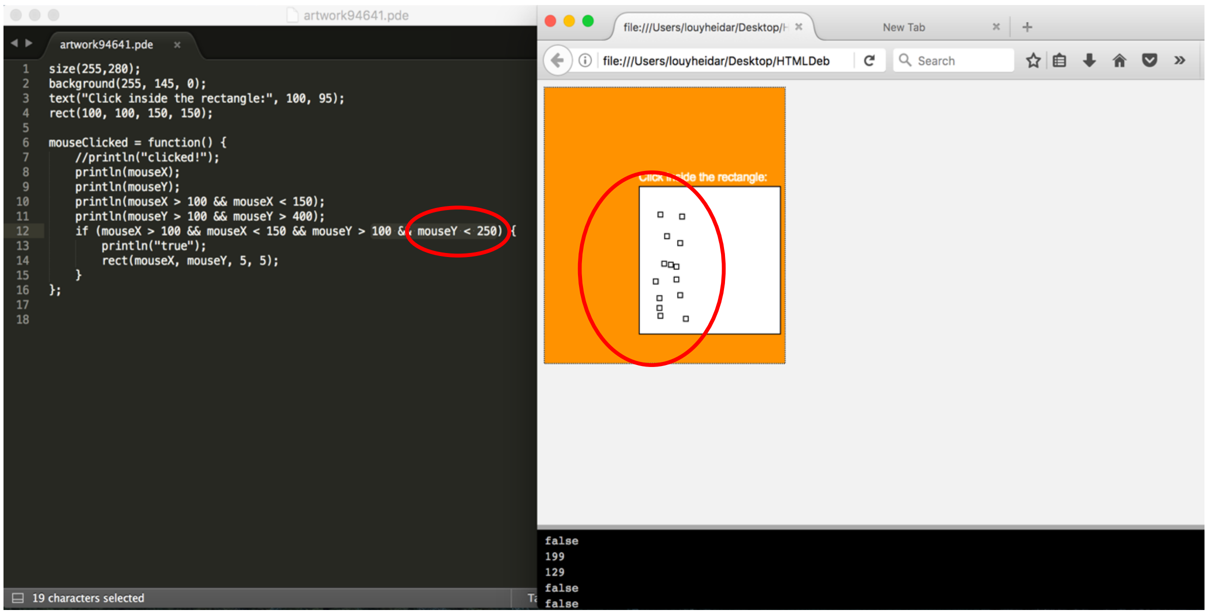


```
size(255,280);
background(255, 145, 0);
text("Click inside the rectangle:", 100, 95);
rect(100, 100, 150, 150);

mouseClicked = function() {
```

Correct Output

100 + 150
mouseY < 250

White rectangles responds to user input on mouse click on the left side but not the right side.


artwork94641.pde
file:///Users/louyheidar/Desktop/H
New Tab
file:///Users/louyheidar/Desktop/HTMLDeb
Search
19 characters selected

```
size(255,280);
background(255, 145, 0);
text("Click inside the rectangle:", 100, 95);
rect(100, 100, 150, 150);

mouseClicked = function() {
    //println("clicked!");
    println(mouseX);
    println(mouseY);
    println(mouseX > 100 && mouseX < 150);
    println(mouseY > 100 && mouseY > 400);
    if (mouseX > 100 && mouseX < 150 && mouseY > 100 && mouseY < 250) {
        println("true");
        rect(mouseX, mouseY, 5, 5);
    }
};
```

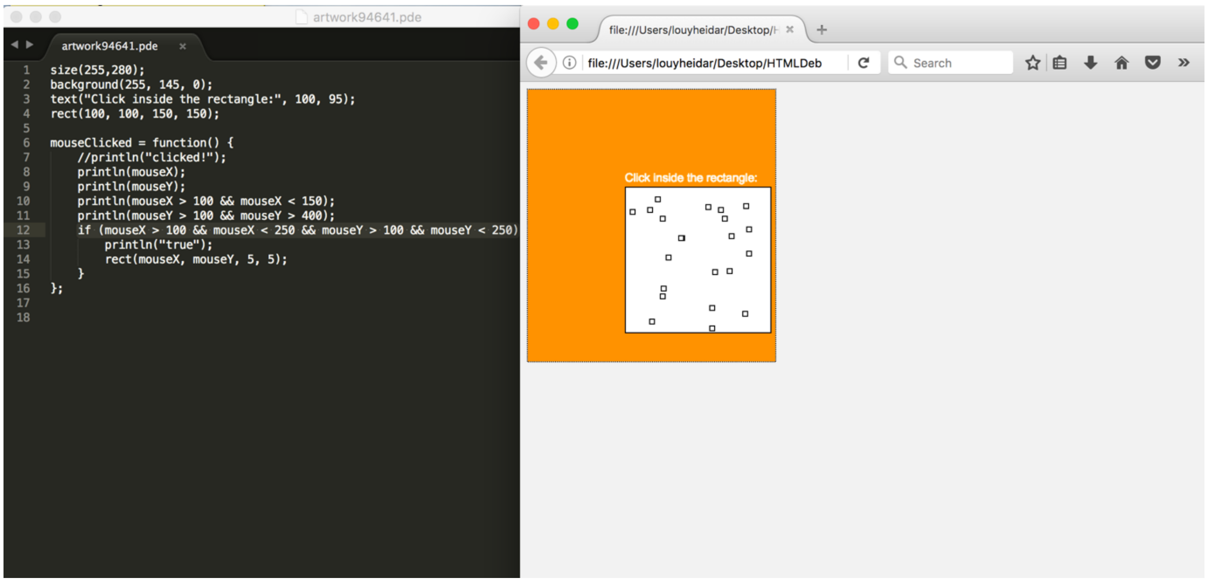
```
false
199
129
false
false
```

Step 7

Apply same concept (Step 6) to the right side of rectangle

Correct Output

if (mouseX > 100 && mouseX < 250 && mouseY > 100 && mouseY < 250)



```
size(255,280);
background(255, 145, 0);
text("Click inside the rectangle:", 100, 95);
rect(100, 100, 150, 150);

mouseClicked = function() {
    //println("clicked!");
    println(mouseX);
    println(mouseY);
    println(mouseX > 100 && mouseX < 150);
    println(mouseY > 100 && mouseY > 400);
    if (mouseX > 100 && mouseX < 250 && mouseY > 100 && mouseY < 250)
        println("true");
        rect(mouseX, mouseY, 5, 5);
    }
};
```

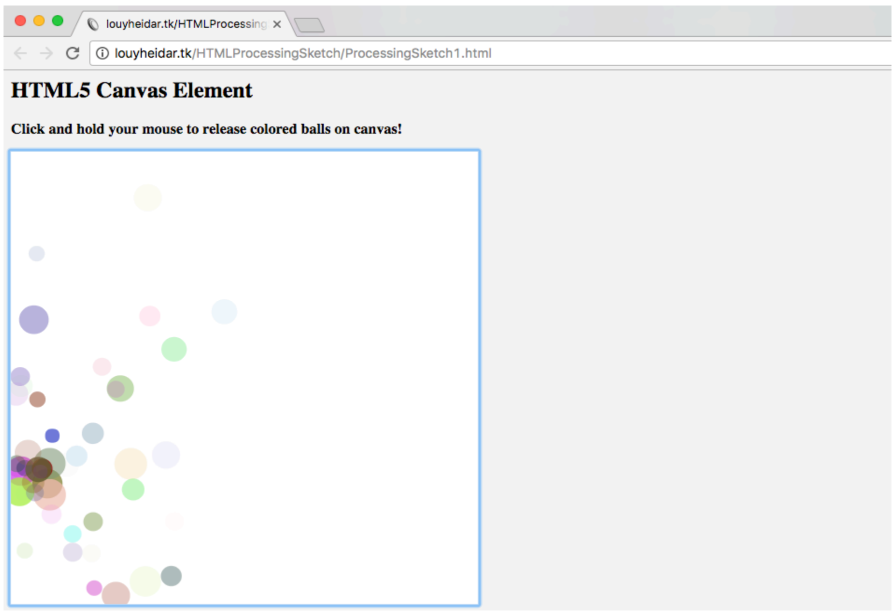
artwork94641.pde
file:///Users/louyheidar/Desktop/HTMLDeb
Search
Click inside the rectangle:

HTML5 Canvas Element

Processing.JS has become relatively common to share your Processing programs on the web. The process works by compiling your Java code into JavaScript and then rendering the output using the HTML5 canvas element in the browser.

Result is a web-based version of your sketch that is rendered without any plugins. This means that your program will be accessible in any modern browser, including mobile devices.

http://louyheidar.tk/HTMLProcessingSketch/ProcessingSketch1.html



louyheidar.tk/HTMLProcessing
louyheidar.tk/HTMLProcessingSketch/ProcessingSketch1.html

HTML5 Canvas Element

Click and hold your mouse to release colored balls on canvas!