**A Lunch Vending Machine Using JAVA**

**By**

**Angela Marks**

**Introduction**

I’m creating a random lunch vending machine using the Java programming language. The most exciting part of this vending machine is that it randomly chooses lunch for the buyer. Sometimes, the buyer has no idea or is too rushed to think about what to have for lunch and wants to make a quick decision; this machine will help the buyer choose one of the numbers provided by the vending machines. I will use two Java packages: Random and Scanner. Once the machine is operating, it will prompt the buyer to put in the number; in this case, it’s just a number one. After that, the program will run through a list of lunches to choose randomly, and the output will be the name of the lunch the buyer will get. The buyer can exit the program by pressing the number zero.

**Creating Scenario**

**Scenario 1:** The Buyer inputs the number provided by the lunch vending machine and gets the food.

1. The lunch vending machine prompts the buyer to input the number “1” to buy food or “0” to exit

the program.

2. The buyer inputs the number “1” to buy lunch.

3. The vending machine starts choosing the lunch menu randomly.

4. The lunch vending machine displays the result of the lunch the buyer will get.

5. The buyer chooses another lunch or type zero to exit the program.

**Scenario 2:** Buyer inputs the number not provided by the lunch vending machine.

1. The lunch vending machine prompts the buyer to input the number “1” to buy food or “0” to exit the program.

2. The buyer inputs the number “2” to buy lunch.

3. The vending machine outputs a statement, “Invalid Choice. Please press 1 or 2”.

4. The buyer chooses the lunch again or types zero to exit the program.

**Pseudocode**

**---------Start the Lunch Vending Machine--------**

*// Initialize the list of lunches*

Set lunches to [ “Ramen”, “Burger”, “Sushi”, “Chicken Salad”, “Sandwiches”]

*// Create a random number generator*

Initialize randomGenerator

*// Display a welcome messages and prompts a buyer to press the number ‘1’ or ‘0’*

Display “Welcome to the Random Lunch Vending Machine!”

Display “Press ‘1’ to get a random lunch, or ‘0’ to exit.”//

*// Start the loopDis*

While True

Display “You choice: “ *// Prompt a buyer to press the number ‘1’ or ‘0’*

Read buyerChoice from Scanner // Get buyer’s input

If buyerChoice is 1 then

*// Generate a random index to select a lunch*

Set randomIndex to random(lunches.length)

//Display the selected lunch

Display “ Your lunch: “ + selectLunch

Else if buyerChoice is 0 then

*// Exit the program*

Display “Thank you for using the lunch vending machine!”

Break

Else

*// Handle invalid input*

Display “Invalid choice. Please press ‘1’ or ‘0’.”

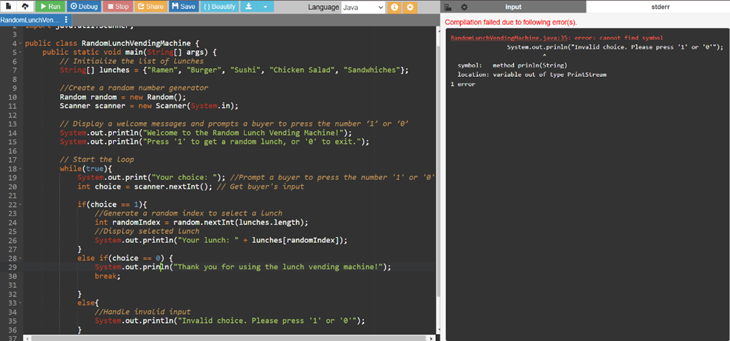
**-----END-----**

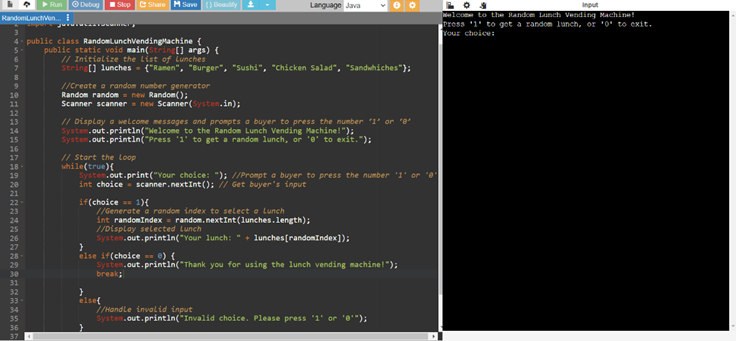
**Testing**

When I started figuring out what project I wanted to create, I came up with the idea of the “Vending Machine.” I researched and looked for some examples online, and I think I want to make it a bit more fun using the Random package. Still, the code is straightforward, so I don’t make many mistakes since I have learned all of them from the course.

1.Typo from Println to Prinln

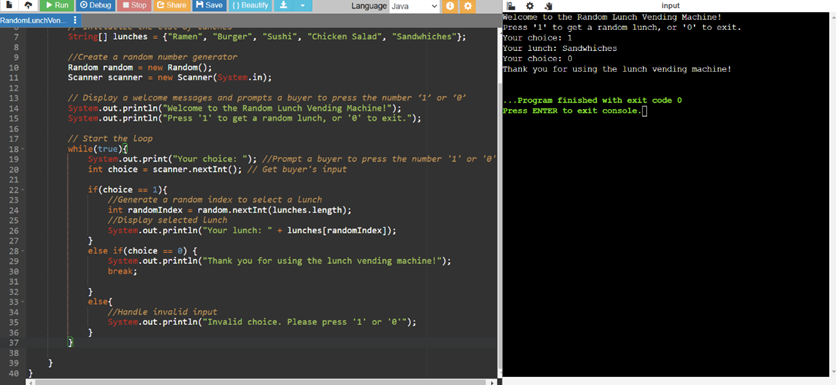
After completing the writing of the code, I tested it for the first time, and it showed an error message “Cannot find symbol” on line 35. I fixed it right away and reran it, it worked perfectly.

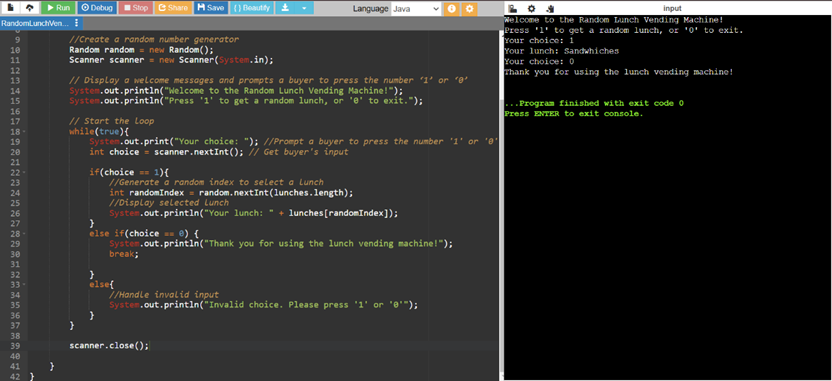




2. Closing the scanner

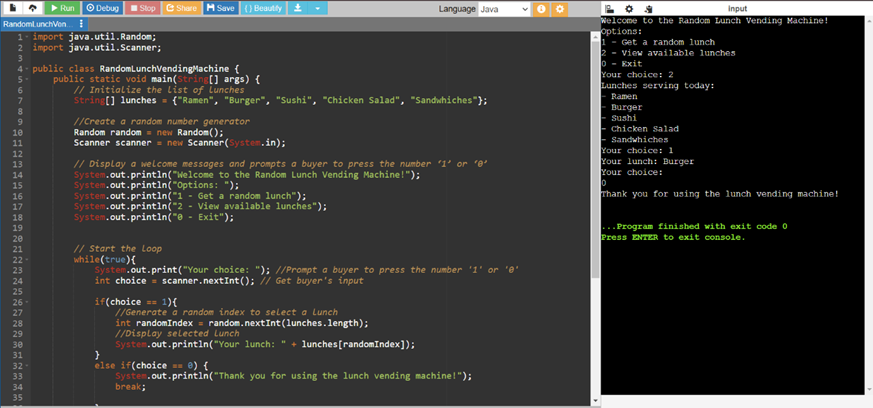
I completed writing the random lunch vending machine without closing the scanner. When running the code, it worked. But, because I did some research about creating the vending machine using Java programming language, I saw there are some people using scanner.close to the end of the code, so I researched more to find out that failing to close the Scanner, especially in the programming like the vending machine might cause some issues when JVM will clean up resources upon program termination. So, I decided to close the scanner to make sure there would be no further issues with the program.





3. Changed the display message to provide more options for the buyer.

In the original version of the random lunch vending machine, the buyer can only choose ‘1’ for the machine to select their lunch randomly or choose ‘0’ to exit the program. I felt like it might be a good idea if the buyer could see a list of the lunches provided in the vending machine before they start to make the machine randomize their lunch selection. So, I decided to create another option for the buyer to see what lunches are available in the vending machine.



**Programming**

import java.util.Random;

import java.util.Scanner;

public class RandomLunchVendingMachine {

public static void main(String[] args) {

// Initialize the list of lunches

String[] lunches = {"Ramen", "Burger", "Sushi", "Chicken Salad", "Sandwhiches"};

//Create a random number generator

Random random = new Random();

Scanner scanner = new Scanner(System.in);

// Display a welcome messages and prompts a buyer to press the number ‘1’, '2' or ‘0’

System.out.println("Welcome to the Random Lunch Vending Machine!");

System.out.println("Options: ");

System.out.println("1 - Get a random lunch");

System.out.println("2 - View available lunches");

System.out.println("0 - Exit");

// Start the loop

while(true){

System.out.print("Your choice: "); //Prompt a buyer to press the number '1' or '0'

int choice = scanner.nextInt(); // Get buyer's input

if(choice == 1){

//Generate a random index to select a lunch

int randomIndex = random.nextInt(lunches.length);

//Display selected lunch

System.out.println("Your lunch: " + lunches[randomIndex]);

}

else if(choice == 0) {

System.out.println("Thank you for using the lunch vending machine!");

break;

}

else if(choice == 2) {

System.out.println("Lunches serving today: ");

for(String lunch: lunches) {

System.out.println("- " + lunch);

}

}

else{

//Handle invalid input

System.out.println("Invalid choice. Please press '1', '2', or '0'");

}

}

scanner.close();

}

}

**Complete Program**

Link to completed program: <https://onlinegdb.com/ZS_y33yTB>