**MODULE 2**

**SECTION 6**

**Problem 1:**

**Validate a Bank PIN**

Overview  
Develop a java program to validate bank PIN of a customer. Use a while loop to repeat code until a valid PIN is entered.  
Task  
1. Declare a valid integer PIN.  
2. Prompt the user to enter the PIN.  
3. In a while loop, perform the following steps:  
• Compare the user-entered PIN with the already declared PIN  
• If the entered PIN is not the same, prompt the user to enter the PIN again  
• Repeat the loop until the correct PIN is entered  
4. Print a message confirming that the correct PIN has been entered and that the user now has access to their account.  
The ValidatePin.java file is available to help you get started.

**ANSWER:**

import java.util.Scanner;

public class ValidatePin {

public static void main(String[] args) {

// Declare a valid integer PIN

final int VALID\_PIN = 1234;

// Create a Scanner object to read user input

Scanner scanner = new Scanner(System.in);

int enteredPin = -1; // Initialize with a value that is not equal to the VALID\_PIN

// Prompt the user to enter the PIN

System.out.println("Please enter your PIN:");

// Use a while loop to keep prompting the user until the correct PIN is entered

while (enteredPin != VALID\_PIN) {

// Read the entered PIN

enteredPin = scanner.nextInt();

// Compare the entered PIN with the valid PIN

if (enteredPin != VALID\_PIN) {

// If the entered PIN is not correct, prompt the user to try again

System.out.println("Incorrect PIN. Please try again:");

}

}

// Close the scanner

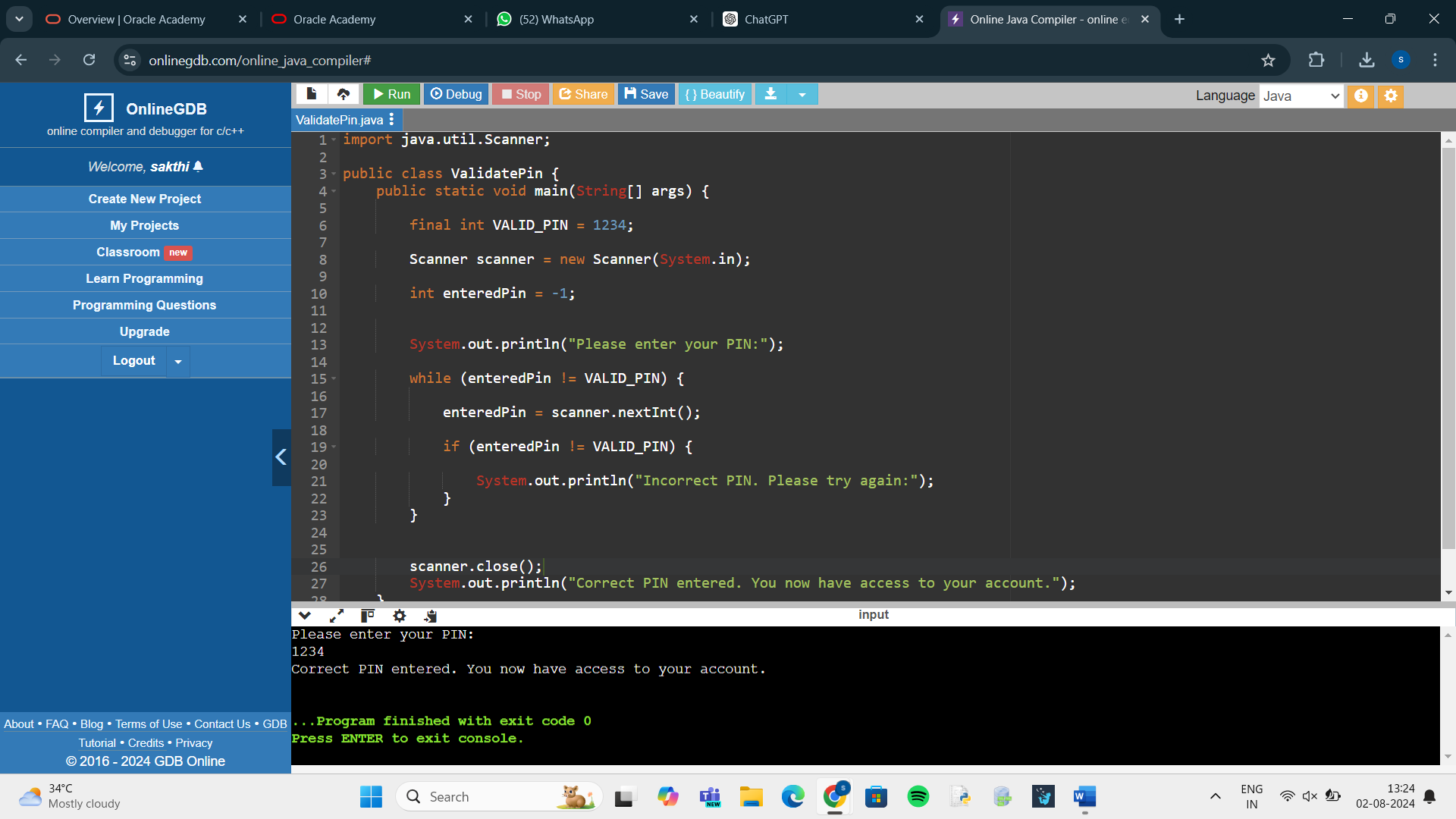
scanner.close();

// Print a message confirming that the correct PIN has been entered

System.out.println("Correct PIN entered. You now have access to your account.");

}

}



**Problem 2:**

**Displaying Multiples of a Number**

Overview  
Develop a java program to calculate the multiples of a given number using a for loop.  
Task  
Have the user enter a number, and then use a for loop to display all the multiples of that number from 1 to 12.  
Expected Output:  
Choose a number: 7  
7x1 = 7  
7x2 = 14  
7x3 = 21  
7x4 = 28  
7x5 = 35  
7x6 = 42  
7x7 = 49  
7x8 = 56  
7x9 = 63  
7x10 = 70  
7x11 = 77  
7x12 = 84

The DisplayMultiples.java file is available to help you get started.

**ANSWER:**

import java.util.Scanner;

public class DisplayMultiples {

public static void main(String[] args) {

// Create a Scanner object to read user input

Scanner scanner = new Scanner(System.in);

// Prompt the user to enter a number

System.out.print("Choose a number: ");

int number = scanner.nextInt();

// Use a for loop to display all the multiples of the entered number from 1 to 12

for (int i = 1; i <= 12; i++) {

System.out.println(number + "x" + i + " = " + (number \* i));

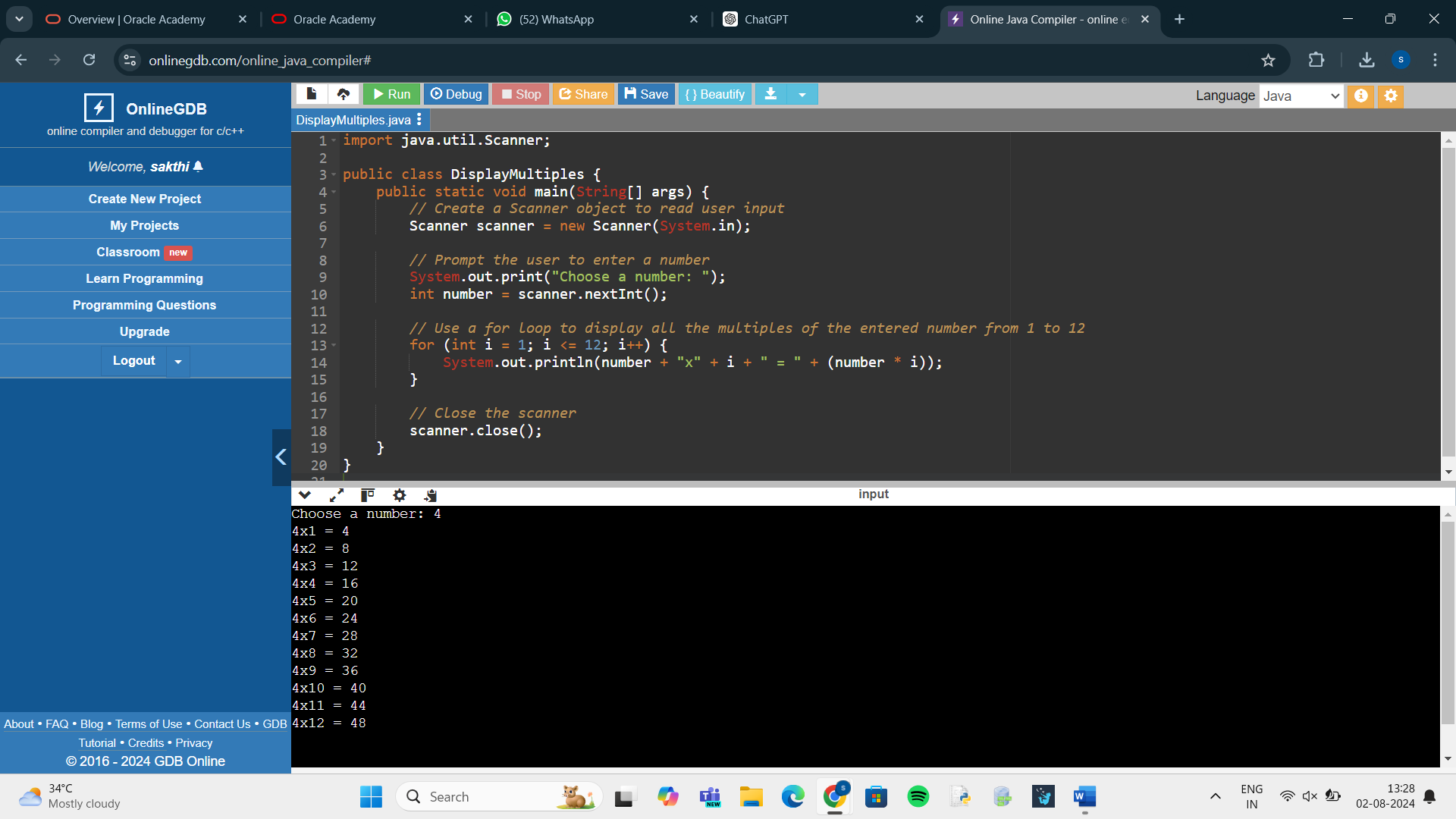
}

// Close the scanner

scanner.close();

}

}



**Problem 3:**

**Programmatic ASCII Art**

Overview  
Using text to create a picture is known as ASCII art. In section 2, we made an ASCII art cat. This required us to type every character in  
the art we wanted to create. In this practice, you’ll find a way to draw basic shapes programmatically in customizable sizes.  
5x4 Rectangle  
5x5 Isosceles Right Triangle  
Task  
Complete the following two methods in LoopShape.java:  
• createRectangle(): This method accepts two arguments for width and height which should be used to print a rectangle  
• createTriangle(): This method accepts one argument for the size of a leg, which should be used to print an isosceles  
right triangle  
Try changing the value of the arguments you’re supplying these two methods from the main method. Make sure your program can  
successfully draw each shape to a custom size. Additionally, your program must:  
• Refuse to draw shapes with any dimension less than 1  
• Be able to draw shapes with any dimension equal to 1 (a 1x1 shape should print just a single character)  
If the problem seems difficult, remember to break it into smaller challenges such as:  
• How do I print a single line that is a variable number of “#” characters wide?  
• How do I create a String that begins and ends with a “#”, but has a variable number of spaces in between?  
Finishing each smaller challenge is an accomplishment. This problem is as much about understanding loops as it’s about  
understanding how to break a big problem into smaller tasks.  
The knowledge you’ve gained in this section on loops will be very helpful in completing this program. You’re free to use whichever type  
of loop statements you feel would be best. You’ll also need to remember a few concepts from previous sections.  
The LoopShape.java and LoopShapeTest.java files are available to help you get started.

**ANSWER:**

public class LoopShape {

public static void main(String[] args) {

// Test the createRectangle method

createRectangle(5, 4);

// Test the createTriangle method

createTriangle(5);

}

// Method to create a rectangle

public static void createRectangle(int width, int height) {

// Check for invalid dimensions

if (width < 1 || height < 1) {

System.out.println("Invalid dimensions. Both width and height must be greater than 0.");

return;

}

// Loop through each row

for (int i = 0; i < height; i++) {

// Loop through each column in the row

for (int j = 0; j < width; j++) {

System.out.print("#");

}

System.out.println();

}

}

// Method to create an isosceles right triangle

public static void createTriangle(int leg) {

// Check for invalid dimension

if (leg < 1) {

System.out.println("Invalid dimension. Leg length must be greater than 0.");

return;

}

// Loop through each row

for (int i = 1; i <= leg; i++) {

// Loop through each column in the row

for (int j = 0; j < i; j++) {

System.out.print("#");

}

System.out.println();

}

}

}

