**Code:**

import java.util.Scanner;

class Main {

public static void main(String[] args) {

int input;

char response;

Scanner scanner = new Scanner(System.in);

System.out.println("\*\*\* Java Times Tables Program \*\*\*");

System.out.println("Please enter the number you wish to see times tables for:");

do {

input = scanner.nextInt();

while (input < 2) {

System.out.println("Error: Input value cannot be lower than 2.");

System.out.println("Please enter the number you wish to see times tables for:");

input = scanner.nextInt();

}

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

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

}

System.out.println("Would you like to view another Times Table? (Y) / (N)");

response = scanner.next().charAt(0);

if (response == 'Y' || response == 'y') {

System.out.println("Please enter the number you wish to see times tables for:");

} else {

System.out.println("Closing program...");

}

} while (response == 'Y' || response == 'y');

}

}

**Explanation:**

– Line #1 calls the ‘scanner’ function.

– Lines #5 and #6 define the variables ‘input’ and ‘response’ as an int and char, respectively.

– Line #7 creates a new scanner called ‘scanner’.

– Line #9 prompts the user to input the number they wish to see the times tables for.

– Line #10 begins a do-while loop, that concludes on Line #27. The program will run once, and then will repeat until the variable ‘response’ is set to the char ‘y’ or ‘Y’ (on Line #21).

– Line #11 sets the value of the variable ‘input’ to the users input into the scanner.

– Line #12 to Line #16 is a while loop. While the value of ‘input’ is less than 2, an error code is printed and the user in prompted to re-enter their number, which resets the value of input.

– Line #17 to Line #19 is a for loop:

(

A: int ‘I’ is defined as 1,

B: while ‘I’ is less than or equal to 12,

C: increase the value of ‘I’ by 1.

This loop will repeat 12 times, printing the times tables by multiplying the value of ‘I’ by the value of ‘input’.

)

– Line #20 asks the user if they wish to view another times table, and prompts them to enter a corresponding character.

– Line #21 sets the value of the variable ‘response’ to the users input into the scanner.

– Line #22 to Line #26 is an if-statement, checking to see if the value of ‘response’ is equal to ‘Y’ or ‘y’. If it is, then the user is prompted to enter another number. If not, a closing program message is printed.

– Line #27 concludes the initial do-while loop. If the value of response is ‘Y’ or ‘y’ to repeat the program, it returns to Line #11 to reset the value of ‘input’. From here, it is validated using the while loop and the new times table is printed, using the new value of input.

**Note:** Use a do-while loop if you want the contents of the loop to run at least once, or a while loop if they do not need to run when unnecessary.