9. Integer Palindrome

Problem: Write a Java program to check if a given integer is a palindrome.

Test Cases:

Input: 121

Output: true

Input: -121

Output: false

EXPLANATION  
Explanation

Your code checks if a number is a palindrome, meaning it reads the same backward as forward. Here’s how it works:

1. Class Declaration: You have two classes - Palindrome and Question9.

2. Palindrome Class:

- Attributes:

- iReversed: Stores the reversed number.

- iUser: Stores the user-input number.

- Constructor: Initializes iUser with the given number.

- Method Duplicate: Reverses the number:

- Extracts the last digit of iNum using modulo (%) and adds it to iReversed after shifting iReversed one place to the left (multiplying by 10).

- Removes the last digit from iNum by integer division by 10.

- Method Check: Compares iReversed with iUser to see if they are the same. Returns true if they are, otherwise false.

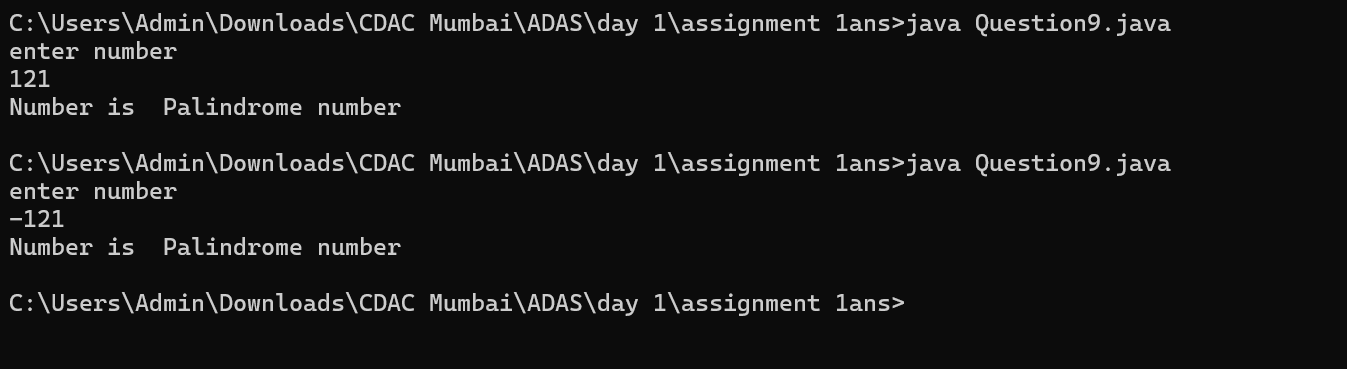
3. Question9 Class:

- Main Method: This is the entry point of your program.

- User Input: It reads an integer iNo from the user.

- Object Creation: Creates an object pobj of the Palindrome class.

- Method Call: Calls Duplicate to reverse the number and Check to see if the reversed number is the same as the original.

- Conditional Output: Prints whether the number is a palindrome based on the result from Check.  
  
  
  
flowchart  
 Start

|

v

Read input `iNo`

|

v

Create `Palindrome` object (`pobj`)

|

v

Call `Duplicate` method

|

v

Initialize `iNum = iUser`

|

v

`iNum != 0` ?

| |

No Yes

| |

v v

End Extract last digit `iDigit = iNum % 10`

|

v

Update reversed `iReversed = iReversed \* 10 + iDigit`

|

v

Remove last digit `iNum = iNum / 10`

|

v

Loop back to `iNum != 0`

|

v

Call `Check` method

|

v

`iReversed == iUser` ?

| |

Yes No

| |

v v

Print "Palindrome" Print "Not Palindrome"

|

v

End