

Object Oriented Programming Using - JAVA

PRACTICAL – 3(A)

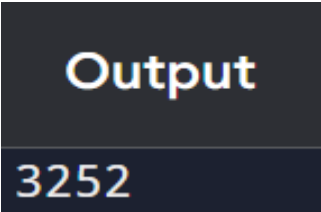
Aim: Write a program in Java to reverse the digits of a number using while Loop

Theory: The reverse number program in Java works by repeatedly extracting the last digit of a given number and constructing the reversed number step by step. It uses the modulus operator % to obtain the last digit and the division operator / to remove that digit from the original number. The extracted digit is then added to a new variable reversed after multiplying the existing value of reversed by 10, which shifts its digits to the left. This process continues inside a while loop until the original number becomes zero. Finally, the value stored in reversed is printed, which represents the reverse of the given number. For example, if the input is 2523, the output will be 3252.

Code:

```
public class reverse
{
    public static void main(String[] args)
    {
        int num = 2523, reversed = 0;
        while (num != 0)
        {
            int digit = num % 10;
            reversed = reversed * 10 + digit;
            num /= 10;
        }
        System.out.println(reversed);
    }
}
```

Output:

A screenshot of a terminal window with a dark background. The word "Output" is displayed in a large, white, sans-serif font. Below it, the number "3252" is displayed in a smaller, white, sans-serif font.

Conclusion: The reverse number program is a simple way to understand how we can break a number into digits and rearrange them. By taking out the last digit again and again and adding it to a new number, we finally get the reverse of the original one. This program makes it easy to learn how loops and small calculations can be used to solve bigger problems in programming.

Object Oriented Programming Using - JAVA

PRACTICAL – 3(B)

Aim: Write a program in Java to reverse the digits of a number using do-while Loop

Theory: The reverse number program in Java works by repeatedly extracting the last digit of a given number and constructing the reversed number step by step. It uses the modulus operator % to obtain the last digit and the division operator / to remove that digit from the original number. The extracted digit is then added to a new variable reversed after multiplying the existing value of reversed by 10, which shifts its digits to the left. This process continues inside a while loop until the original number becomes zero. Finally, the value stored in reversed is printed, which represents the reverse of the given number. For example, if the input is 5678, the output will be 8765.

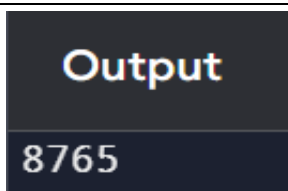
Code:

```
public class reverse
{
    public static void main(String[] args)
    {
        int num = 5678, reversed = 0;

        do
        {
            int digit = num % 10;
            reversed = reversed * 10 + digit;
            num /= 10;
        }
        while (num != 0);

        System.out.println(reversed);
    }
}
```

Output:



Conclusion: The reverse number program is a simple way to understand how we can break a number into digits and rearrange them. By taking out the last digit again and again and adding it to a new number, we finally get the reverse of the original one. This program makes it easy to learn how loops and small calculations can be used to solve bigger problems in programming.