▨ DAY 6 – While Loop

Class Programs → 4

Assignment Programs → 4

☆ 1. What is a While Loop?

A while loop is used when we don’t know exactly how many times the loop must run, but a condition controls it.

Syntax

```
while (condition) {
   statements;
}
```

Important Points:

Condition is checked before the loop runs.

If the condition is false initially → loop will not execute even once.

Must update the variable inside the loop to avoid infinite loop.

==

☆ ☆ CLASS PROGRAM – 1

Program: Print numbers from 1 to 10 using while loop

==

Pseudo Code

```
Start
i = 1
```

```
While i <= 10:
    Print i
    i = i + 1
End
```

Flow

1. Start with i = 1

2. Check condition → i ≤ 10

3. Print i

4. Increase i

5. Loop continues

Variables

i → loop counter

Program

```
class WhileLoop1 {
    public static void main(String args[]) {

        int i = 1;

        while (i <= 10) {
            System.out.println(i);
            i++;
        }
    }
}
```

Output

```
1
2
3
4
5
6
7
8
9
10
```

==

⭐ ⭐ CLASS PROGRAM – 2

Program: Print even numbers from 1 to 20

==

Pseudo Code

```
Start
i = 1
While i <= 20:
   If i % 2 == 0:
      Print i
   i = i + 1
End
```

Flow

Loop from 1 to 20

Check even numbers

Print only even

Variables

i → increases 1 each time

Program

```
class EvenUsingWhile {
    public static void main(String args[]) {

        int i = 1;

        while (i <= 20) {

            if (i % 2 == 0)
                System.out.println(i);

            i++;
        }
    }
}
```

Output

```
2
4
6
...
20
```

==

⭐ ⭐ CLASS PROGRAM – 3

Program: Print sum of numbers from 1 to 10

==

Pseudo Code

```
Start
i = 1
```

```
sum = 0
While i <= 10:
    sum = sum + i
    i = i + 1
Print sum
End
```

Flow

Start sum = 0

Add numbers 1 to 10

Print total sum

Variables

i → loop counter

sum → stores running total

Program

```
class SumUsingWhile {
    public static void main(String args[]) {

        int i = 1, sum = 0;

        while (i <= 10) {
            sum += i;
            i++;
        }

        System.out.println("Sum = " + sum);
    }
}
```

Output

```
Sum = 55
```

==

⭑ ⭑ CLASS PROGRAM – 4

Program: Print digits of a number (e.g., 1234 → 1 2 3 4)

==

Pseudo Code

```
Start
Take n
While n > 0:
    digit = n % 10
    Print digit
    n = n / 10
End
```

Flow

Extract last digit

Print it

Remove last digit

Continue

Variables

n → input number

digit → stores last digit extracted

Program

```
class DigitsWhile {
    public static void main(String args[]) {
        int n = 1234;

        while (n > 0) {
            int digit = n % 10;
            System.out.println(digit);
            n = n / 10;
        }
    }
}
```

Output

```
4
3
2
1
```

(If you want correct order → use reverse logic later.)

==

⭐⭐⭐ ASSIGNMENT PROGRAMS – 4

==

⭐ Assignment – 1

Program: Print numbers from 20 to 1 (reverse)

```
class ReverseWhile {
    public static void main(String args[]) {

        int i = 20;

        while (i >= 1) {
            System.out.println(i);
            i--;
        }
    }
}
```

Output

```
20
19
18
...
1
```

⭐ Assignment – 2

Program: Count digits of a number (ex: 4567 → 4 digits)

```
class CountDigits {
    public static void main(String args[]) {

        int n = 4567;
        int count = 0;

        while (n > 0) {
            count++;
            n = n / 10;
        }

        System.out.println("Count = " + count);
    }
}
```

Output

```
Count = 4
```

⭐ Assignment – 3

Program: Reverse a number (ex: 123 → 321)

```
class ReverseNumber {
    public static void main(String args[]) {

        int n = 123;
        int rev = 0;

        while (n > 0) {
            int digit = n % 10;
            rev = rev * 10 + digit;
            n = n / 10;
        }

        System.out.println("Reversed = " + rev);
    }
}
```

Output

```
Reversed = 321
```

⭑ Assignment – 4

Program: Print table of a number using while loop (example: 8)

```
class TableWhile {
    public static void main(String args[]) {

        int i = 1;

        while (i <= 10) {
            System.out.println("8 x " + i + " = " + (8 * i));
            i++;
        }
    }
}
```

Output

```
8 x 1 = 8
8 x 2 = 16
...
8 x 10 = 80
```