

SWT11022: Practical for Fundamentals of Programming
Department of Information & Communication Technology Faculty
of Technology
South Eastern University of Sri Lanka

Time: - 09.30 am – 12.30 pm

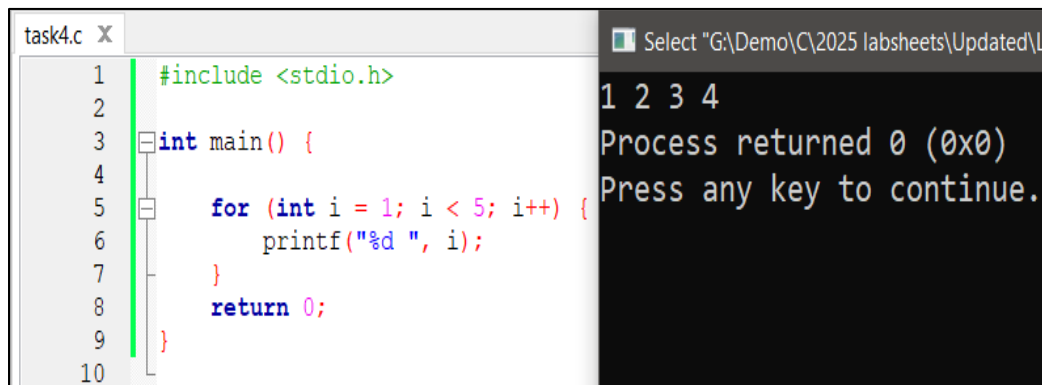
Lab Sheet 07

Title: Introduction to the Loop**Objective:**

- Understand and practice the for loop.
- Understand and practice the while loop.
- Understand and practice the do-while loop.
- Understand and practice loop control statements.

Practical 1: for Loop**Steps:****1. for Loop:**

- Use a for loop to print the numbers from 1 to 5.
- Initialize the loop control variable, set the condition, and specify the increment.



The image shows a code editor window titled 'task4.c' on the left and a terminal window on the right. The code in the editor is a C program that uses a for loop to print numbers from 1 to 5. The terminal shows the output of the program, which is the numbers 1, 2, 3, and 4, followed by the message 'Process returned 0 (0x0)' and 'Press any key to continue.'.

```
task4.c X
1  #include <stdio.h>
2
3  int main() {
4
5      for (int i = 1; i < 5; i++) {
6          printf("%d ", i);
7      }
8      return 0;
9  }
10
```

Select "G:\Demo\C\2025 labsheets\Updated\L
1 2 3 4
Process returned 0 (0x0)
Press any key to continue.

Practical 2: while Loop

Steps:

1. while Loop:

- Use a while loop to print even numbers from 2 to 10.
- Set up a condition to check if a number is even.

The screenshot shows a C program in a text editor and its execution output. The code in `task4.c` is as follows:

```

1  #include <stdio.h>
2
3  int main() {
4      int number = 2;
5
6      while (number <= 10) {
7          printf("%d ", number);
8          number += 2;
9      }
10
11     return 0;
12 }
13

```

The output window shows the execution results:

```

2 4 6 8 10
Process returned 0 (0x0)
Press any key to continue.

```

Practical 3: do-while Loop

Steps:

1. do-while Loop:

- Use a do-while loop to ask the user for a positive number.
- Continue asking until a positive number is provided. int input;

The screenshot shows a C program in a text editor and its execution output. The code in `task4.c` is as follows:

```

1  #include <stdio.h>
2
3  int main() {
4      int input;
5
6      do {
7          printf("Enter a positive number: ");
8          scanf("%d", &input);
9      } while (input <= 0);
10
11     printf("You entered a positive number: %d\n", input);
12
13     return 0;
14 }
15

```

The output window shows the execution results:

```

Enter a positive number: -3
Enter a positive number: 0
Enter a positive number: 12
You entered a positive number: 12

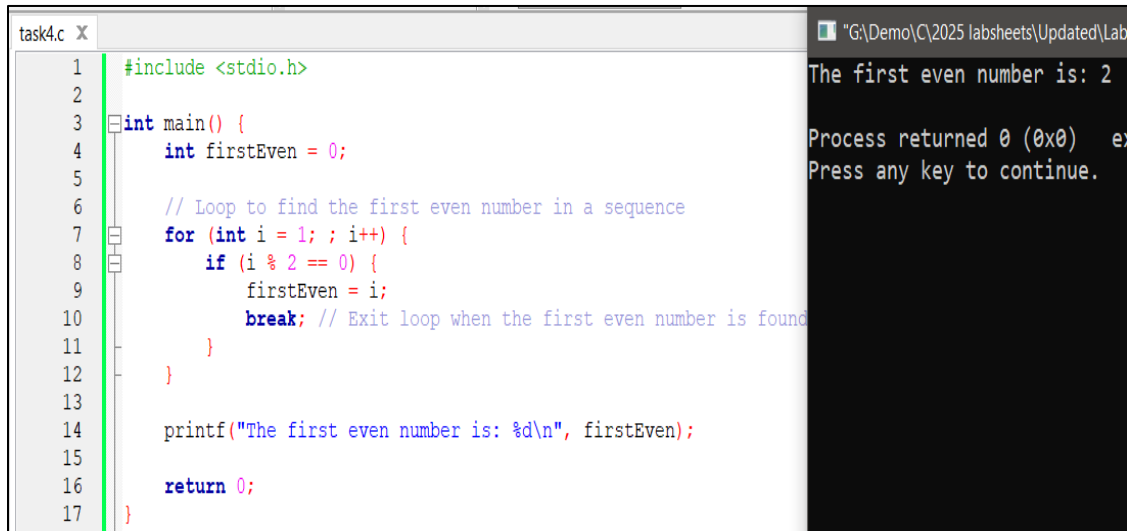
Process returned 0 (0x0)  executio
Press any key to continue.

```

Practical 4: Loop Control Statements

1. Using break Statement:

- Use a for loop to find the first even number in a sequence.
- Use the break statement to exit the loop when an even number is found.



The screenshot shows a C program in a text editor and its execution output. The code defines a variable `firstEven` and a `for` loop starting from `i = 1`. Inside the loop, it checks if `i` is even (`i % 2 == 0`). If true, it assigns `firstEven = i` and uses `break;` to exit the loop immediately. The output shows the first even number found is 2.

```

1  #include <stdio.h>
2
3  int main() {
4      int firstEven = 0;
5
6      // Loop to find the first even number in a sequence
7      for (int i = 1; ; i++) {
8          if (i % 2 == 0) {
9              firstEven = i;
10             break; // Exit loop when the first even number is found
11         }
12     }
13
14     printf("The first even number is: %d\n", firstEven);
15
16     return 0;
17 }

```

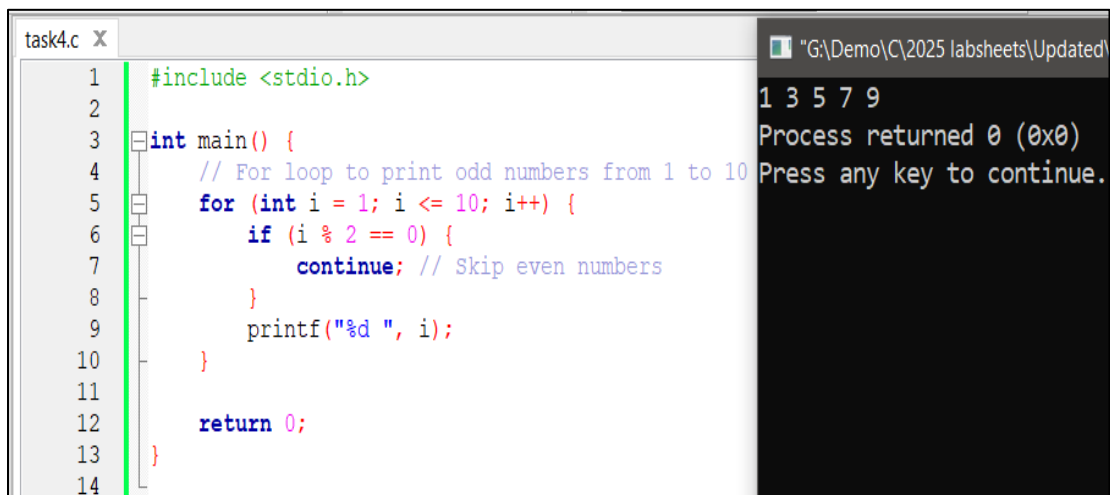
Output: The first even number is: 2

Process returned 0 (0x0) e)

Press any key to continue.

2. Using continue Statement:

- Use a for loop to print odd numbers in a sequence.
- Use the continue statement to skip even numbers.



The screenshot shows a C program in a text editor and its execution output. The code uses a `for` loop from `i = 1` to `i = 10`. Inside the loop, it checks if `i` is even (`i % 2 == 0`). If true, it uses `continue;` to skip the rest of the loop body and move to the next iteration. If false, it prints the value of `i`. The output shows only odd numbers from 1 to 9.

```

1  #include <stdio.h>
2
3  int main() {
4      // For loop to print odd numbers from 1 to 10
5      for (int i = 1; i <= 10; i++) {
6          if (i % 2 == 0) {
7              continue; // Skip even numbers
8          }
9          printf("%d ", i);
10     }
11
12     return 0;
13 }
14

```

Output: 1 3 5 7 9

Process returned 0 (0x0)

Press any key to continue.

Tasks

1. Write a C program to print all alphabets from a to z using for loop.
2. Write a C program to print all odd number between 1 to 100 using while loop.
3. Write a C program to print natural numbers from 1 to 10 using do...while loop.
4. Write a C program that uses a while loop to continuously take input from the user until the user enters a negative number. Use the break statement to exit the loop when the user enters a negative number.
5. Write a C program to print all even numbers from 1 to 20 using a for loop. Use the continue statement to skip the odd numbers.

Report Submission Guidelines

- Submit the Report by 24/03/2025.
- Late submissions will not be accepted.
 - Report Structure Practical No
 - Date of Submission
 - Title
 - Objective of the practical.
 - Exercise
 - Challenges
 - Conclusion
 - References