CONTROL STRUCTURES

QUICK EXERCISE

- Predict the output:

```
for(i=1;i<=5;i++)
{
for(j=1;j<=4;j++)
{
printf(“*”);
}
printf(“\n”);
}
```

Output:

```
****
****
****
****
****
```

QUICK EXERCISE

- **Predict the output:**

```
for(i=1;i<=5;i++)
{
for(j=1;j<=4;j++)
{
printf(“%d”,i);
}
printf(“\n”);
}
```

Output:

```
1111
2222
3333
4444
5555
```

QUICK EXERCISE

- **Predict the output:**

```
for(i=1;i<=5;i++)
{
for(j=1;j<=4;j++)
{
printf(“%d”,j);
}
printf(“\n”);
}
```

Output:

```
1234
1234
1234
1234
1234
```

QUICK EXERCISE

- Predict the output:

```
for(i=5;i>=1;i--)
{
for(j=1;j<=4;j++)
{
printf(“%d”,i);
}
printf(“\n”);
}
```

Output:

```
5555
4444
3333
2222
1111
```

QUICK EXERCISE

- Predict the output:

```
for(i=1;i<=5;i++)
{
for(j=4;j>=1;j--)
{
printf("%d",j);
}
printf("\n");
}
```

Output:

```
4321
4321
4321
4321
4321
```

QUICK EXERCISE

- Print the following pattern for ‘n’ rows:

```
*
**
***
****
*****
```

QUICK EXERCISE

- Print the following pattern for ‘n’ rows:

1

1 2

1 2 3 4

1 2 3 4 5

……

1 2 3 4 5 …..n

QUICK EXERCISE

- Print the following pattern for ‘n’ rows:

```
*****
****
***
**
*
```

QUICK EXERCISE

- Print the following pattern for ‘n’ rows:

```
A
AB
ABC
ABCD
ABCDE
```

QUICK EXERCISE

- Print the following pattern for ‘n’ rows:

```
    *
   **
  ***
 ****
*****
```

JUMP STATEMENTS

- The jump statements can skip a set of statements and take the control elsewhere in the program.
- There are 4 types of jump statements in C
  - break
  - continue
  - goto
  - return

JUMP STATEMENTS

Keyword	Where they can be placed	Where they take the control
break	• switch • loops	• outside the switch • outside the loop (by abruptly breaking the loop)

continue	loops	To the next iteration/cycle i.e
		for – updation
		while – condition
		do… while - condition

goto	Anywhere in the program	Statement with the corresponding label
return	In functions	Outside the function

BREAK

- Example:

```
for( i=1;i<=5;i++)
{
if ( i==3 )
   break;
else
   printf (“%d”,i);
}
```

Output:

```
1 2
```

CONTINUE

- Example:

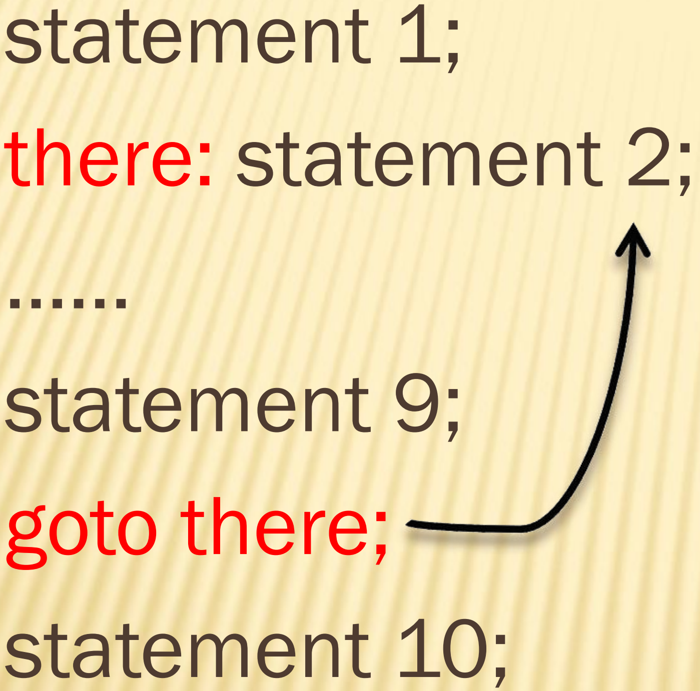
```
for( i=1;i<=5;i++)
{
if ( i==3 )
   continue;
else
   printf (“%d”,i);
}
```

Output:

```
1 2 4 5
```

GOTO

- Syntax:



```
statement 1;
there: statement 2;
……
statement 9;
goto there;
statement 10;
```

QUICK EXERCISE

- Predict the output:

```
for ( i=1;i<=4;i++)
{
for(j=1;j<=4;j++)
{
if(i==j)
break;
printf(“i=%d  j=%d\n”,i,j);
}
}
```

Output:

```
i=2  j=1
i=3  j=1
i=3  j=2
i=4  j=1
i=4  j=2
i=4  j=3
```

QUICK EXERCISE

- Predict the output:

```
for ( i=1;i<=4;i++)
{
for(j=1;j<=4;j++)
{
if(i==j)
continue;
printf(“i=%d  j=%d\n”,i,j);
}
}
```

QUICK EXERCISE

- Predict the output (from previous slide):

```
for ( i=1;i<=4;i++)
{
for(j=1;j<=4;j++)
{
if(i==j)
continue;
printf(“i=%d  j=%d\n”,i,j);
}
}
```

Output:

```
i=1  j=2
i=1  j=3
i=1  j=4
i=2  j=1
i=2  j=3
i=2  j=4
i=3  j=1
i=3  j=2
i=3  j=4
i=4  j=1
i=4  j=2
i=4  j=3
```

QUICK EXERCISE

- Predict the output:

```
int i=1;
while(i<=10)
{
if(i>=3&&i<=6)
{
i++;
continue;
}
printf(" Hello %d\n",i);
i++;
}
```

Output:

```
Hello  1
Hello  2
Hello  7
Hello  8
Hello  9
Hello  10
```

QUICK EXERCISE

- Predict the output:

```
int i=1;
condition: if(i<=5)
{
printf("Hello %d\n", i);
i++;
goto condition;
}
```

Output:

```
Hello 1
Hello 2
Hello 3
Hello 4
Hello 5
```

QUICK EXERCISE

- Print the following pattern using break keyword:

```
*
**
***
****
*****
```

- Print whether a number is prime or composite.