Conditional Statements and Loops in C

In C programming, break, goto, and continue statements are control flow mechanisms that alter the execution sequence of a program. Let's explore each of these statements in detail:

1. Break Statement

The break statement is used to exit a loop or switch statement prematurely, meaning it immediately terminates the loop or switch and transfers control to the statement that follows the loop or switch.

Usage in Loops:

- Example (in a loop):

```c
for (int i = 1; i <= 10; i++) {
    if (i == 5) {
        break; // Exit the loop when i equals 5
    }
    printf("%d\n", i);
}
```

Output:

```
1
2
3
4
```

```c
    case 1:
        printf("Monday\n");
        break;
    default:
        printf("Invalid day\n");
}
```

Output:

```
Wednesday
```

Course & Test Series
CBSE
ICSE
NTSE
Banking & Insurance
Central Govt. Service
State Govt. Services
LAW Entrance
MBA Entrance
Railways & Metro Services
...many more
abhyasonline.in

2. Goto Statement

The goto statement provides a way to transfer control to another part of the program. It's generally considered bad practice to use goto as it can make code harder to follow and maintain. However, in some cases, it might be useful for jumping to a specific section of code, like breaking out of multiple nested loops.

Usage:

- Example:

```c
int i = 1;
start:
printf("%d\n", i);
i++;
if (i <= 5) {
    goto start; // Jump back to the label 'start'
}
```

Output:

```
1
2
3
4
5
```

3. Continue Statement

The continue statement skips the current iteration of a loop and moves control back to the beginning of the loop for the next iteration. It does not terminate the loop but skips the remaining code inside the loop for the current iteration.

Course & Test Series
CBSE
ICSE
NTSE
Banking & Insurance
Central Govt. Service
State Govt. Services
LAW Entrance
MBA Entrance
Railways & Metro Services
...many more
abhyasonline.in

Introduction to 'C' Language - Module 6

Usage In Loops:

- Example:

```c
for (int i = 1; i <= 5; i++) {
    if (i == 3) {
        continue; // Skip the rest of the loop when i equals 3
    }
    printf("%d\n", i);
}
```

Output:

```
1
2
4
5
```

Solved Example of Break Statement

`break` Statement

Example 1: Exiting a Loop

```c
#include <stdio.h>

int main() {
    for (int i = 1; i <= 10; i++) {
        if (i == 6) {
            break; // Exit the loop when i equals 6
        }
        printf("i = %d\n", i);
    }
    return 0;
}
```

Output:

```
i = 1
i = 2
i = 3
i = 4
i = 5
```

Explanation:

- The `break` statement exits the inner loop when `i * j` equals 4. The outer loop continues with the next iteration.

CBSE
ICSE
NTSE
Banking & Insurance
Central Govt. Service
State Govt. Services
LAW Entrance
MBA Entrance
Railways & Metro Services
...many more
abhyasonline.in

Breaking out of a Nested Loop

```c
#include <stdio.h>

int main() {
    for (int i = 1; i <= 3; i++) {
        for (int j = 1; j <= 3; j++) {
            if (i * j == 4) {
                break; // Exit the inner loop when the product equals 4
            }
            printf("i = %d, j = %d\n", i, j);
        }
    }
    return 0;
}
```

Output:

```
i = 1, j = 1
i = 1, j = 2
i = 1, j = 3
i = 2, j = 1
i = 2, j = 2
i = 3, j = 1
```

Explanation:

- The `break` statement exits the inner loop when `i * j` equals 4. The outer loop continues with the next iteration.

Solved Example of GOTO Command

Simple Usage of `goto`

```c
#include <stdio.h>

int main() {
    int i = 1;

    start:
    printf("%d\n", i);
    i++;
    if (i <= 3) {
        goto start; // Jump back to the label 'start'
    }

    printf("Loop finished\n");
    return 0;
}
```

Output:

```
1
2
3
Loop finished
```

Explanation:

- The `goto` statement sends control back to the `start` label, effectively creating a loop that runs until `i` is greater than 3.

CBSE
ICSE
NTSE
Banking & Insurance
Central Govt. Service
State Govt. Services
LAW Entrance
MBA Entrance
Railways & Metro Services
...many more
abhyasonline.in

Solved Example of Continue Command

Example 1: Skipping an Iteration

```c
#include <stdio.h>

int main() {
    for (int i = 1; i <= 5; i++) {
        if (i == 3) {
            continue; // Skip the rest of the loop when i equals 3
        }
        printf("%d\n", i);
    }
    return 0;
}
```

Output:

```
1
2
4
5
```

Explanation:

- The `continue` statement skips the rest of the loop body when `i` equals 3, so 3 is not printed.

CBSE
ICSE
NTSE
Banking & Insurance
Central Govt. Service
State Govt. Services
LAW Entrance
MBA Entrance
Railways & Metro Services
...many more
abhyasonline.in

Course & Test Series
CBSE
ICSE
NTSE
Banking & Insurance
Central Govt. Service
State Govt. Services
LAW Entrance
MBA Entrance
Railways & Metro Services
...many more
abhyasonline.in

Introduction to 'C' Language - Module 6

Example 2: Continue in a While Loop

```c
#include <stdio.h>

int main() {
    int i = 0;

    while (i < 5) {
        i++;
        if (i == 3) {
            continue; // Skip the rest of the loop when i equals 3
        }
        printf("%d\n", i);
    }
    return 0;
}
```

Output:

```
1
2
4
5
```

Explanation:

- The `continue` statement skips the iteration when `i` equals 3, so 3 is not printed.

Assignment

Ques 1: Print Multiples of 3 from 1to 20 (skip Others) by using Continue statement.
Ques 2: Write a c program to check if a number is positive or negative using a Goto statement.
Ques 3: Write a C program to exit a loop when a number is divisible by 5. Using Break Statement.