Switch-Case & Break

Example 24

```
switch(month) {
    case 1 : printf("January\n");    break;
    case 2 : printf("February\n");   break;
    case 3 : printf("March\n");      break;
    case 4 : printf("April\n");      break;
    case 5 : printf("May\n");        break;
    case 6 : printf("June\n");       break;
    case 7 : printf("July\n");       break;
    case 8 : printf("August\n");     break;
    case 9 : printf("September\n"); break;
    case 10: printf("October\n");    break;
    case 11: printf("November\n");   break;
    case 12: printf("December\n");   break;
    default: printf("No such month\n");
             // Break is not need here
}
```

Switch-Case & Break

Example 25

```
#include <stdio.h>
int main() {
    int n;

    printf("Enter the telephone code: ");
    scanf("%d", &n);

    printf("Area code   City\n");
    switch (n) {
        case 11:  printf("11          Delhi\n"); break;
        case 22:  printf("22          Mumbai\n"); break;
        case 33:  printf("33          Kolkata\n"); break;
        case 40:  printf("40          Chennai\n"); break;
        default:  printf("Area code is not recognized\n");
                  // break not necessary here
    }
}
```

Switch-Case & Break

Example 26

```
#include <stdio.h>
int main() {
  int n, ndigit = 0;

  printf("Enter a number (<=999): ");
  scanf("%d", &n);

  if (n < 10) ndigit = 1;
  else if (n < 100) ndigit = 2;
       else if (n < 1000) ndigit = 3;

  // Rest of the program
}
```

Switch-Case & Break

Example 26 (contd)

```
switch (ndigit) {
  case 1: printf("Input: %d output: %d\n", n, n);
          break;
  case 2: printf("Input: %d output: %d\n", n, n%10*10 + n/10);
          break;
  case 3: printf("Input: %d output: %d\n", n,
                  (n % 10 * 10 + (n / 10) % 10) * 10 + n / 100);
          break;
 default: printf("Invalid input\n");
}
```

Switch-Case & Break

Example 27

```c
#include <stdio.h>

int main() {
    char o;
    int op1, op2;

    printf("Enter operator(+,-,*,/), q to quit : ");
    scanf("%c", &o);
    if (o == 'q')
            return 0; // Exit from main (terminate program)
    else {
            printf("Enter operands : ");
            scanf("%d %d", &op1, &op2);
    }
    // switch-case statement for the operations
}
```

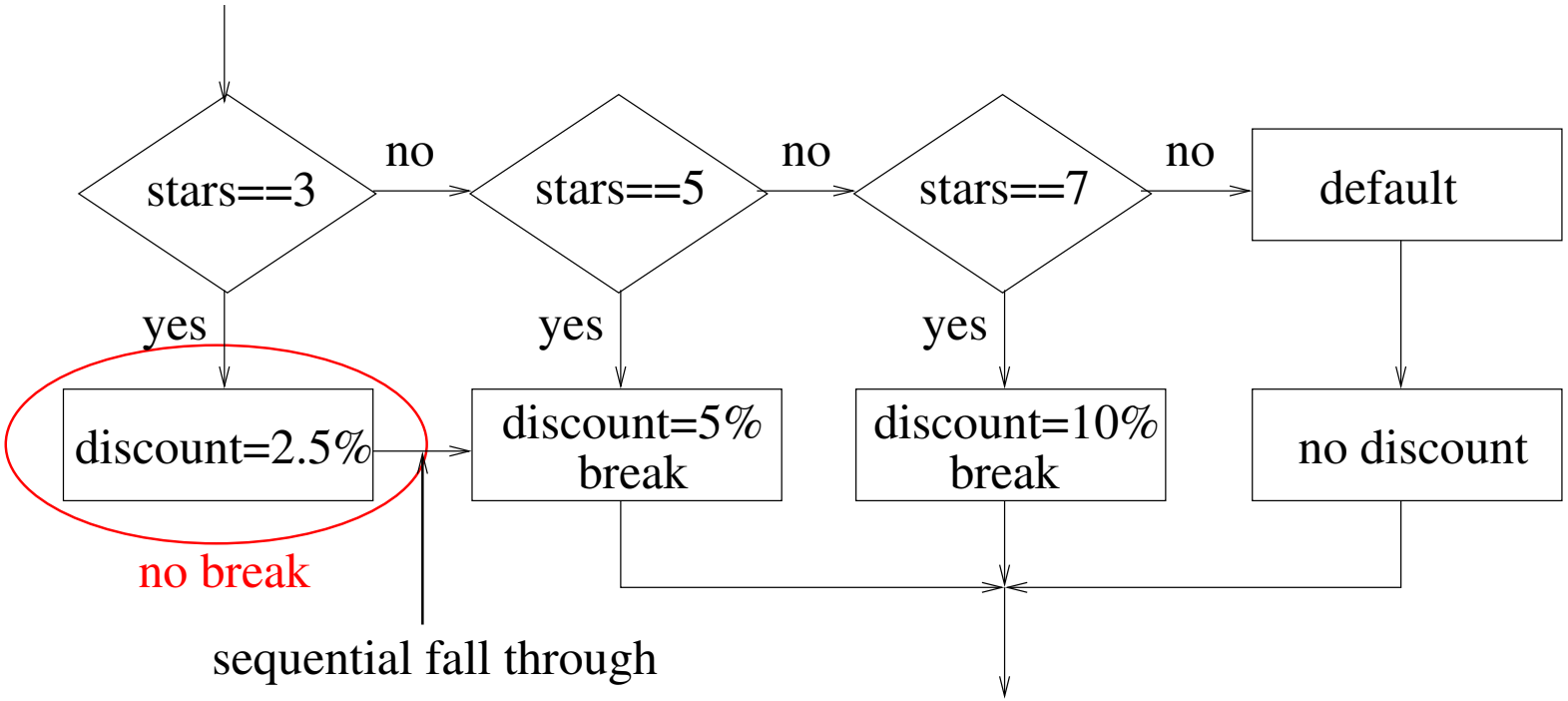
Switch-Case & Break

Example 28 (contd)

```
switch (o) {
    case '+' : printf("%d %c %d = %d\n", op1, o, op2, op1 + op2);
               break;
    case '-' : printf("%d %c %d = %d\n", op1, o, op2, op1 - op2);
               break;
    case '*' : printf("%d %c %d = %d\n", op1, o, op2, op1 * op2);
               break;
    case '/' : if (op2 != 0)
                   printf("%d %c %d = %d\n", op1, o, op2, op1 / op2);
               else
                   printf("Division by 0 not possible\n");
               break;
    default  : printf("not a valid operator\n");
               // break is not needed here
}
```

Switch-Case & Break

Sequential Fall Through



Switch-Case & Break

Example 29

```
#include <stdio.h>
int main() {
    char c;
    scanf("%c", &c);
    switch (c) {
        case '0': // Sequential
        case '1': //   .
        case '2': //   .
        case '3': //   .
        case '4': // Fall
        case '5': //   .
        case '6': //   .
        case '7': //   .
        case '8': // Through
        case '9': printf("%c is a Digit \n", c); break;
        default : printf("%c is not a digit \n", c);
    }
}
```

Loops

Why Loops

- To automate repetition of computation.
- To iterate until the occurrence of an event
- To attempt operation until successful or limit of attempts exceeded.

Loops

Examples of Repetitiveness

- Repetitive computation is a major requirement.
- Arithmetic computation on a sequence:
  - determine next number of the sequence,
  - compute sum or product of the sequence.
- Statistical computation on sequence of numbers:
  - max, min, average, std deviation.

Loops

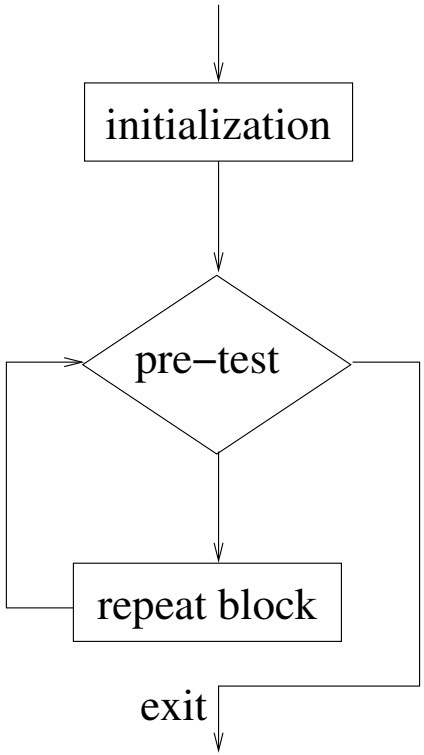
Types of Loops

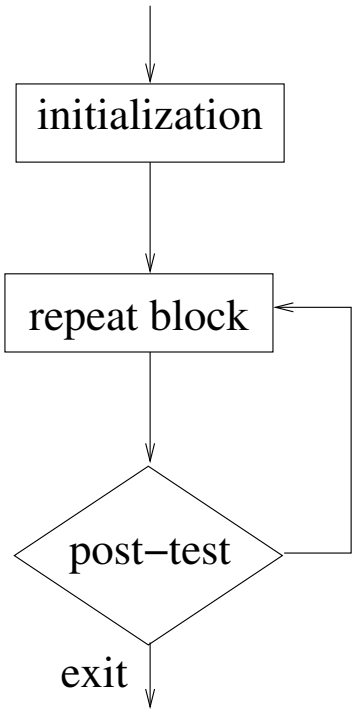
- **Counter controlled loops**: control variable counting up/down (normal loops)
- **Event controlled loops**: until special value is encountered. (E.g., terminate loop when input is 'q')
- **Result controlled loops**: continues until a test determines that the desired result is reached (eg, numerical approximations)

While Loop

Control Flow

```
while (expr) { body }
```





While Loop

Example 30: Counter Controlled While

```
#include <stdio.h>
int main() {
    int i = 0, n;
    double sum=0.0, x;

    printf("Enter number of values to read: ");
    scanf("%d", &n)

    // DON'T FORGET to initialize i before entering loop.
    while (i < n) {
        printf("Enter next value: ");
        scanf("%lf", &x);   // Reading a double
        sum += x;
        i++;                // DON'T FORGET to increment i
    }
    printf("Average of 10 values = %.3f\n", sum/10.0);
}
```