ESC101N
Fundamentals of Computing

Arnab Bhattacharya
arnabb@iitk.ac.in

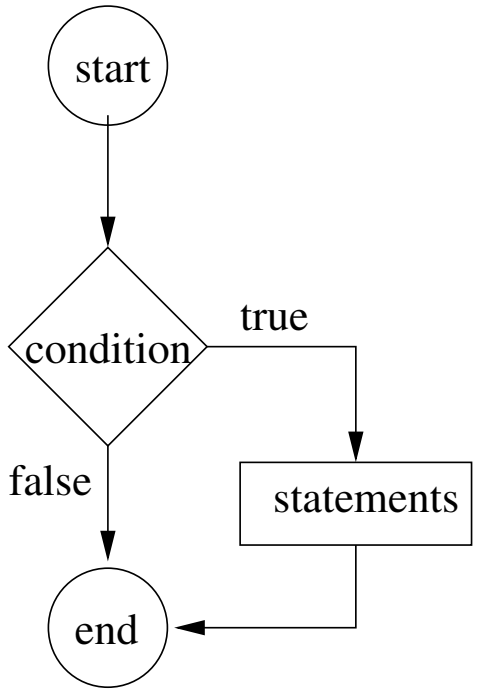
Indian Institute of Technology, Kanpur
http://www.iitk.ac.in/esc101/

1st semester, 2010-11

Tue, Wed, Fri 0800-0900 at L7

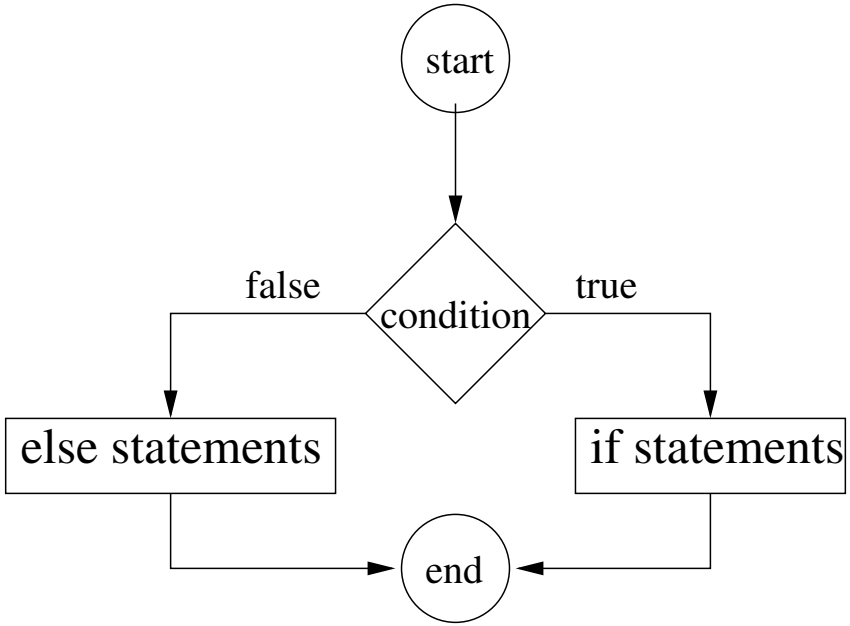
Flow of statements for `if`

```
if (condition)
{
   statements
}
```



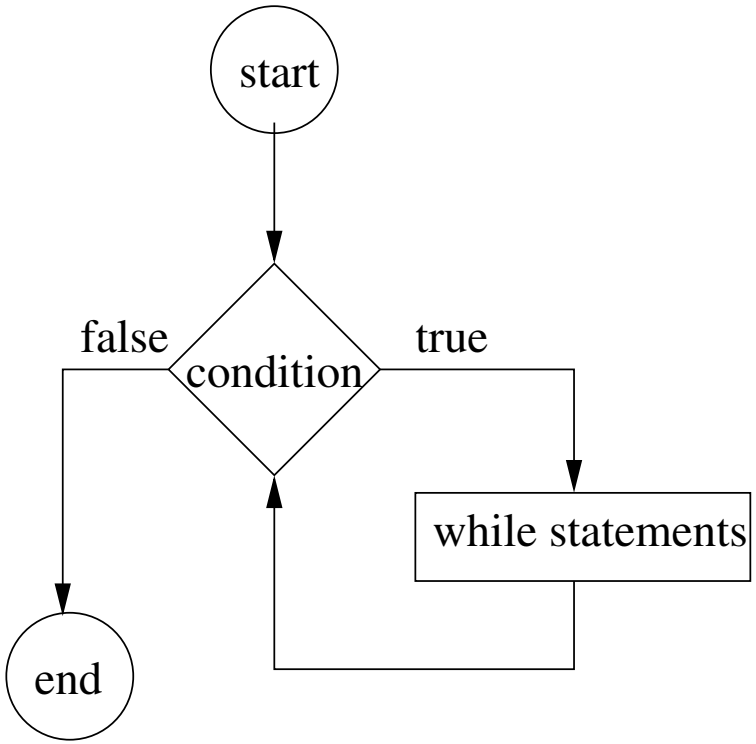
Flow of statements for `if else`

```
if (condition)
{
   statements
}
else
{
   statements
}
```



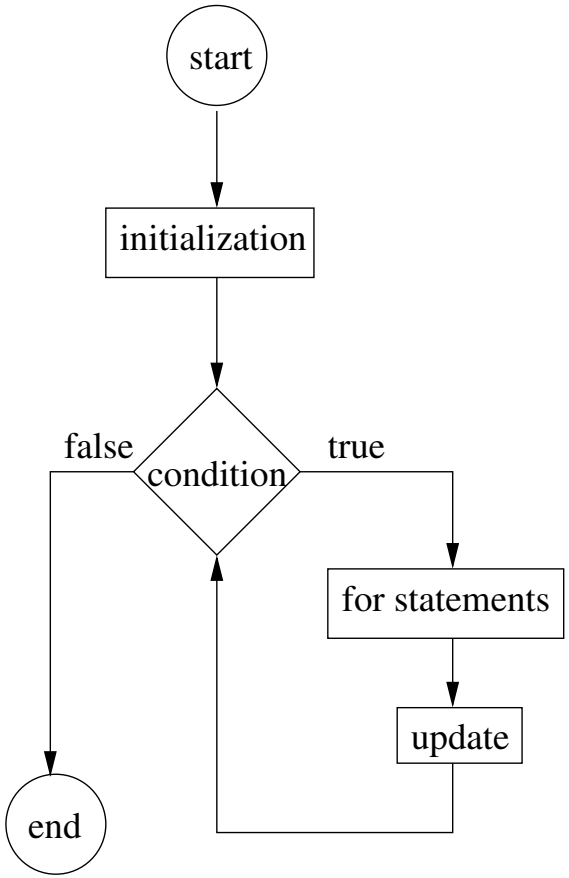
Flow of statements for `while`

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



Flow of statements for `for`

```
for (initialization; condition; update)
{
   statements
}
```



Example: Calculating electricity bill

- Suppose electricity bill is computed according to the following chart:
  - If consumption is less than 100, charge bill at 1%
  - If it is more than 100 but less than 300, charge excess bill at 3% on whatever is in excess of 100
  - If it is more than 300 but less than 800, charge excess bill at 5% on whatever is in excess of 300
  - If it is more than 800, charge excess bill at 7% on whatever is in excess of 800

Algorithm: Calculating electricity bill

- There are 4 intervals based on consumption
  - `consumption <= 100`:
    - Calculate at 1% for consumption
  - `consumption > 100` but `consumption <= 300`:
    - Calculate at 1% for 100 = 1
    - Calculate at 3% for (consumption - 100)
  - `consumption > 300` but `consumption <= 800`:
    - Calculate at 1% for 100 = 1
    - Calculate at 3% for (300 - 100) = 6
    - Calculate at 5% for (consumption - 300)
  - `consumption > 800`:
    - Calculate at 1% for 100 = 1
    - Calculate at 3% for (300 - 100) = 6
    - Calculate at 5% for (800 - 300) = 25
    - Calculate at 7% for (consumption - 800)

Program: Calculating electricity bill

```c
#include <stdio.h>

int main()
{
    double consumption;
    double bill = 0.0;

    printf("Enter consumption: ");
    scanf("%lf", &consumption);

    if (consumption <= 100)
    {
        bill = 0.01 * consumption;
    }
    else if (consumption <= 300)
    {
        bill = 1 + 0.03 * (consumption - 100);
    }
    else if (consumption <= 800)
    {
        bill = 1 + 6 + 0.05 * (consumption - 300);
    }
    else
    {
        bill = 1 + 6 + 25 + 0.07 * (consumption - 800);
    }

    printf("Bill = %lf\n", bill);
}
```

Program: Primality testing

```c
#include <stdio.h>

int main()
{
    int i, n;

    printf("Enter a number for primality testing: ");
    scanf("%d", &n);

    for (i = 2; i <= (n + 1) / 2; i++)
    {
        if ((n % i) == 0)
        {
            break;
        }
    }

    if (((n % i) == 0) && (n != i))
    {
        printf("%d is not a prime\n", n);
    }
    else
    {
        printf("%d is a prime\n", n);
    }
}
```

Program: Primality testing: smart version 1

```c
#include <stdio.h>

int main()
{
    int i, n;

    printf("Enter a number for primality testing: ");
    scanf("%d", &n);

    for (i = 3; i <= n / 2; i = i + 2)
        if ((n % i) == 0)
            break;

    if (((n % i) == 0) && (n != i))
        printf("%d is not a prime\n", n);
    else
        printf("%d is a prime\n", n);
}
```

- Fails for 4

Program: Primality testing: smart version 2

```c
#include <stdio.h>

int main()
{
    int i, n;

    printf("Enter a number for primality testing: ");
    scanf("%d", &n);

    for (i = 3; i <= n / 2; i = i + 2)
        if ((n % i) == 0)
            break;

    if (((n % 2) == 0) || (((n % i) == 0) && (n != i)))
        printf("%d is not a prime\n", n);
    else
        printf("%d is a prime\n", n);
}
```

- Fails for 2

Program: Primality testing: smart version 3

```c
#include <stdio.h>

int main()
{
    int i, n;

    printf("Enter a number for primality testing: ");
    scanf("%d", &n);

    for (i = 3; i <= i / 2; i = i + 2)
        if ((n % i) == 0)
            break;

    if ((n != 2) && (((n % 2) == 0) || (((n % i) == 0) && (n != i))))
        printf("%d is not a prime\n", n);
    else
        printf("%d is a prime\n", n);
}
```

- Perfect

Program: Primality testing with error checking: version 1

```c
#include <stdio.h>

int main()
{
    int i, n;

    printf("Enter a positive integer (greater than 1) for primality testing: ");
    scanf("%d", &n);

    if (n < 2)
    {
        printf("Input number is less than 2\n");
        return 1;
    }

    for (i = 2; i <= (n + 1) / 2; i++)
        if ((n % i) == 0)
            break;

    if (((n % i) == 0) && (n != i))
        printf("%d is not a prime\n", n);
    else
        printf("%d is a prime\n", n);
}
```

- Exits the program when there is an error in the input

Program: Primality testing with error checking: version 2

```c
#include <stdio.h>

int main()
{
    int i, n;

    printf("Enter a positive integer (greater than 1) for primality testing: ");
    scanf("%d", &n);

    while (n < 2)
    {
        printf("Input number is less than 2\n");
        printf("Enter a positive integer (greater than 1) for primality testing: ");
        scanf("%d", &n);
    }

    for (i = 2; i <= (n + 1) / 2; i++)
        if ((n % i) == 0)
            break;

    if (((n % i) == 0) && (n != i))
        printf("%d is not a prime\n", n);
    else
        printf("%d is a prime\n", n);
}
```

- Keeps on asking the user for correct input

Example: Calculating decimal value from binary

- Suppose a binary number is input
  - Input binary digits
  - From the first non-binary digit, ignore the rest
- Convert it into decimal
- Algorithm
  - Initialize number to 0
  - Multiply number by 2 and add the binary digit

Program: Calculating decimal value from binary

```c
#include <stdio.h>

int main()
{
    char c;
    int number = 0;

    scanf("%c", &c);
    while ((c == '0') || (c == '1'))
    {
        number = number * 2 + (c - '0');
        scanf("%c", &c);
    }

    printf("Decimal value is %d\n", number);
}
```