CS 101 Computer Programming and Utilization

Lecture 5
Contol Flow: Iterations, loop design

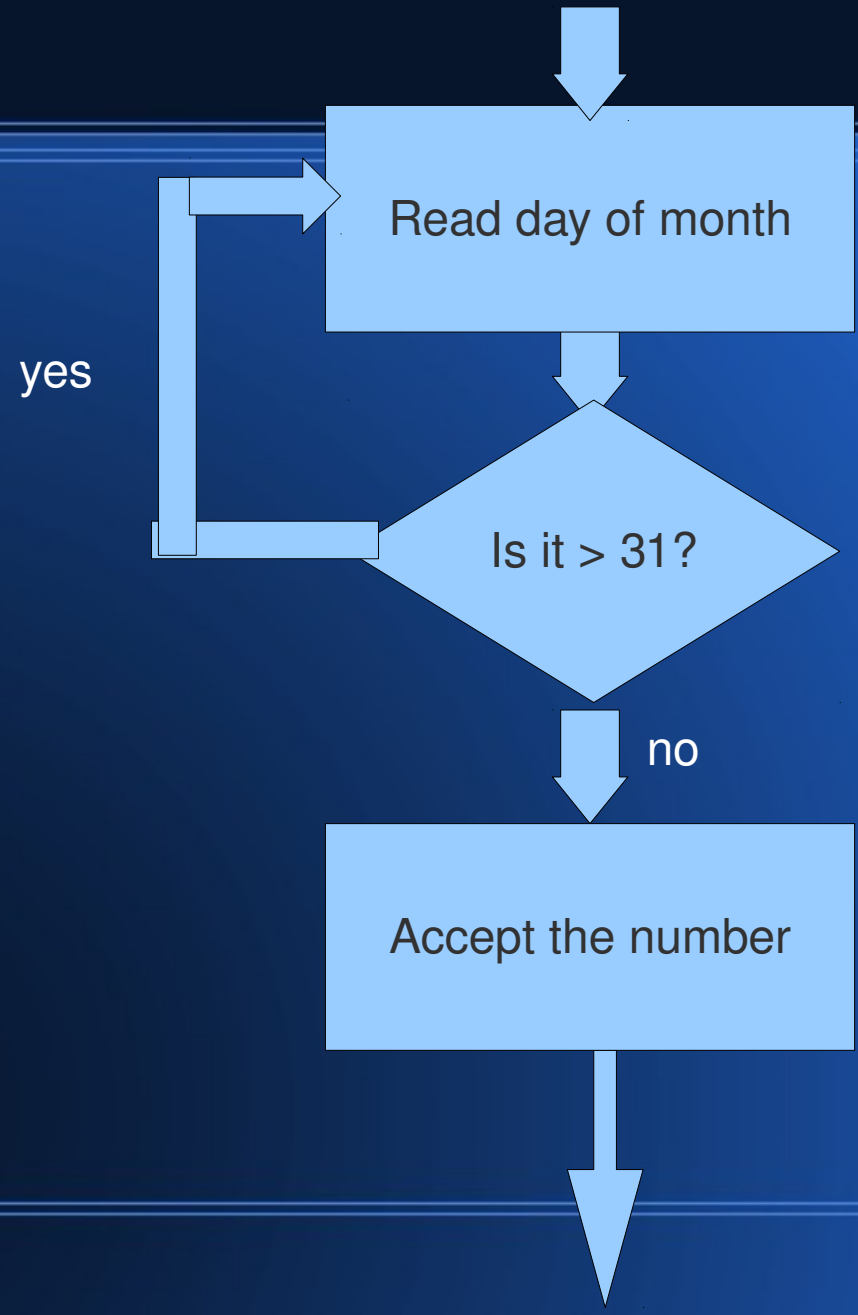
Jan 21 Friday 11:05-12:30 PCS D2
Jan 25 Tuesday 2:00-3:30 FCK D4

Prof. R K Joshi
Computer Science and Engineering
IIT Bombay
Email: rkj@cse.iitb.ac.in

Revision: Storage, Value representation, Control Flow

- Decimal vs. binary numbers
- Binary representations of types such as characters
- Calculating no. of possible values of a type based on storage sizes
- Calculating storage size requirements based on no. of possible values of a given type
- Expressions in assignment
- Every expression returns a value
- The value returned from an expression may or may not be used
- Expressions: comparison, equality, assignment,..
- Sequential control: simple life.. no decisions to make {..;...;...;}
- Branching: if else

Conditional Iteration: do while



Input a day of month
Do not accept if the day provided
  Is > 31
  Instead ask the user
  to retype it.

We have a loop here...

do while statement

do *statement* while (*C*);

Do the statement again and again while C stays true. When c becomes false, the loop will terminate

```
do {
    cin >> day;
} while (day > 31);
cout << “chosen day:” << day << endl;
```

do while statement .. a more complex condition

```
do {
    cin >> day;
} while ((day > 31) ??? (day<1))
cout << “chosen day:” << day << endl;
```

What if you wanted to display an error message every time input is given wrongly?

Where do you accommodate this?

demo..

A solution...
But there is redundancy!

```
do {
    cin >> day;
    if ((day>31)||(day<1)) cout << “invalid input…”;
} while ((day > 31) || (day<1))
cout << “chosen day:” << day << endl;
```

Removing redundancy inside the loop.. But still there is a redundancy problem! Make an improvement?

```
cin >> day;
    if ((day<31)&&(day>1)) {
        cout << “chosen day:” << day << endl;
        return 0;
    }
    else {
        do {
            cout << “invalid input...”;
            cin >> day;
        } while ((day > 31) || (day<1));
        cout << “chosen day:” << day << endl;
    }
```

The condition is repeated, but it's now out of the loop..

```
cin >> day;
 if ((day>31)||(day<1))
    do {
        cout << “invalid input...”;
       cin >> day;
     } while ((day > 31) || (day<1));
else { };
 cout << “chosen day:” << day << endl;
```

Another form: While do statement

Syntax: while (C) statement;

While C stays true, do the statement.

```
cin >> day;
while ((day >31) || (day<1)) {
    cin >> day;
}
cout << “chosen day:” << day << endl;
```

Attempt flow chart for while do?