Programming Language Lab

Bibhas Adhikari

IIT Kharagpur

September 15, 2020

Pseudo code - example

A decision-making problem statement

Allow a customer to deposit or withdraw money from a bank account, and if a user elects to withdraw funds, ensure that sufficient monies exist

Example

```
 if action == deposit
    Deposit funds into account
else
    if balance < withdraw amount
      Insufficient funds for transaction
    else
     Withdraw monies
    end if
end if
```

Flow chart

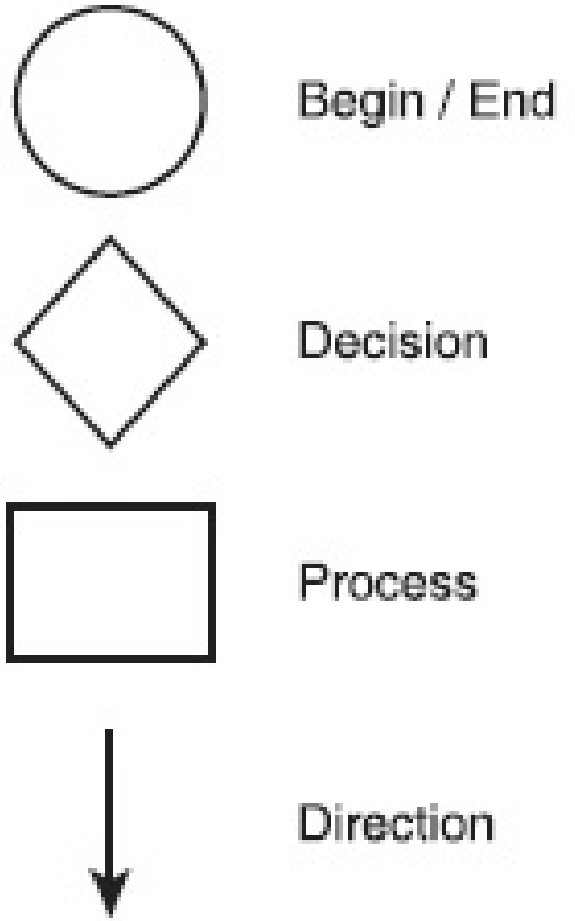

Figure: Flowchart symbols

Flow chart

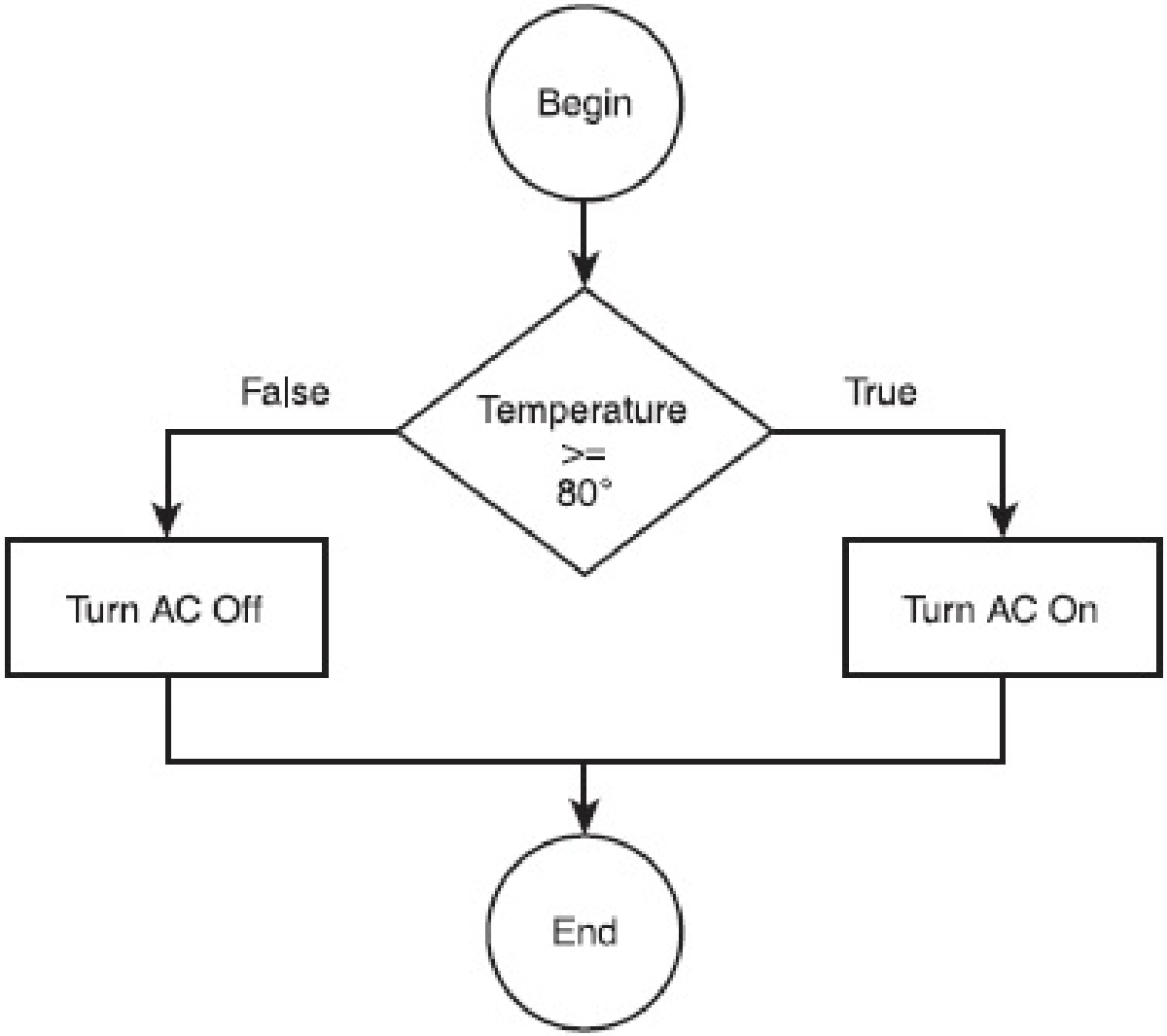

Figure: Flowchart for AC problem

Flow chart

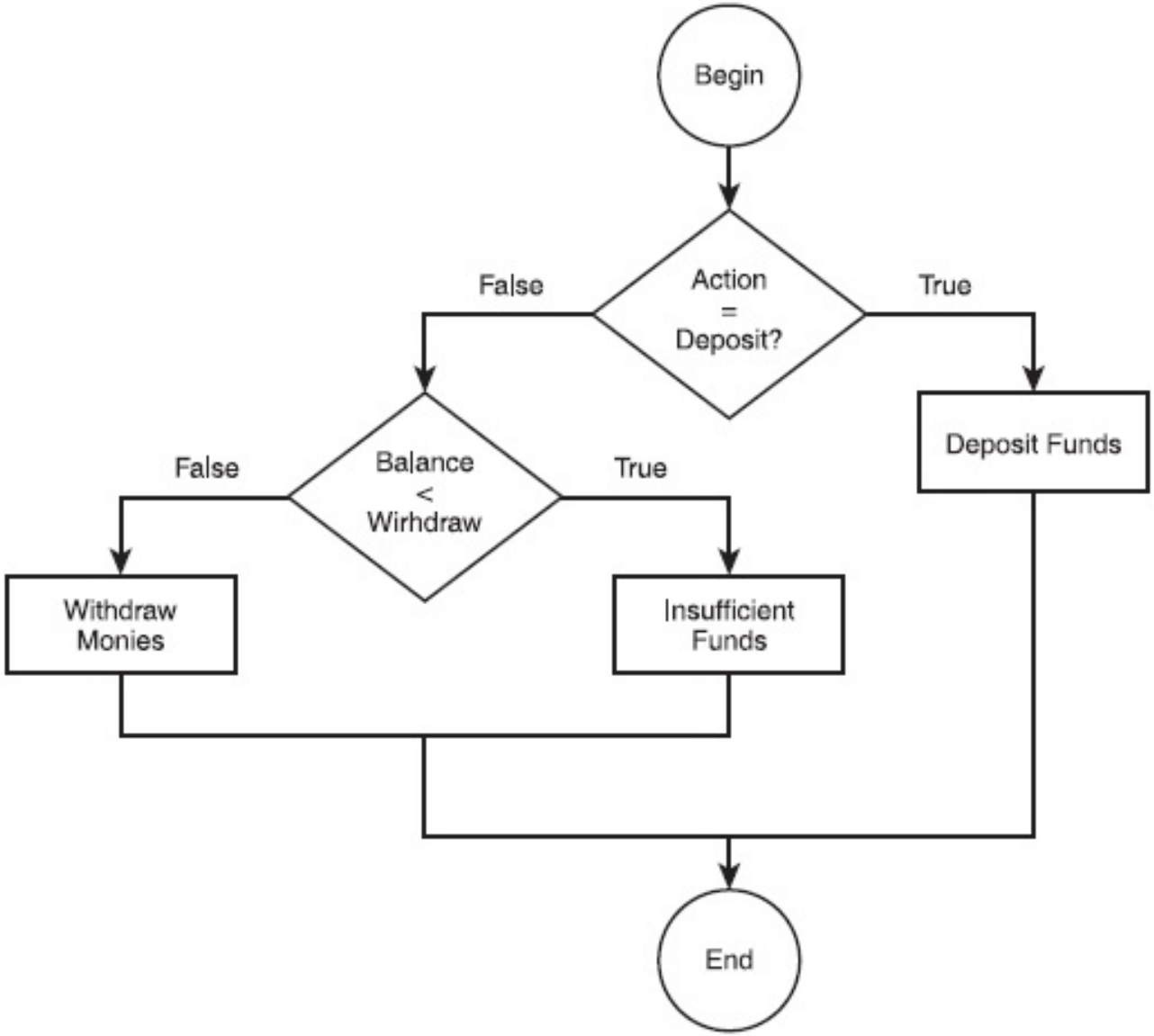

Figure: Flowchart for bank problem

C program for AC problem

```
 if (iTemperature >= 80) {
//Turn AC on
     printf("AC is on");
}
 else {
    //Turn AC off
     printf("\nThe AC is off \n");
}
```

Pseudo code to C program

Problem statement

Allow a customer to deposit or withdraw money from a bank account, and if a user elects to withdraw funds, ensure that sufficient monies exist

Example: Nested *if*

```
 if (action == deposit) {
//deposit funds into account
    printf("\nFunds deposited \n");
}
else {
     if (balance < withdraw)
         //insufficient funds
     else
         //withdraw monies
}
```

Boolean Algebra: compound *if*

Operators in BA

and, *or*, *not*

Truth table for *and*, and *or* operators

X	Y	$X \wedge Y$ (*and*)	$X \vee Y$ (*or*)
true	true	true	true
true	false	false	true
false	true	false	true
false	false	false	false

Example

"$3 = 3$ *and* $4 = 4$", "$1 = 2$ *or* $1 = 1$"

Boolean Algebra: compound if

Truth table for not operators

X	Result (not)
true	false
false	true

Example

" $not\ (2 = 2)$"

Order of operations: Left to right

Let $x = 1, y = 2, z = 3$

- $z < y\ or\ z <= z\ and\ x < z$
- $z < y\ or\ (z < x\ and\ x < z)$

C : compound if

Character set	Boolean operator
&&	and
$\|\|$	or

Example

- $if\ (3 > 1\ \&\&\ 5 < 10)$
- $if\ (3 > 5\ \|\|\ 5 <= 5)$

An application

How to checking whether a particular number belongs in a range of values, an interval using and and or?

Random numbers

- The *rand* function an integer from 0 to a library-defined number, generally at least 32767
- *(rand() % y) + x* can generate a number in the range from x to y
- However, *rand* function produce a number repeatedly in the same interval
- *srand(time(0))* function produces a number randomly (current time as seed of random number generator)

Loop Structure

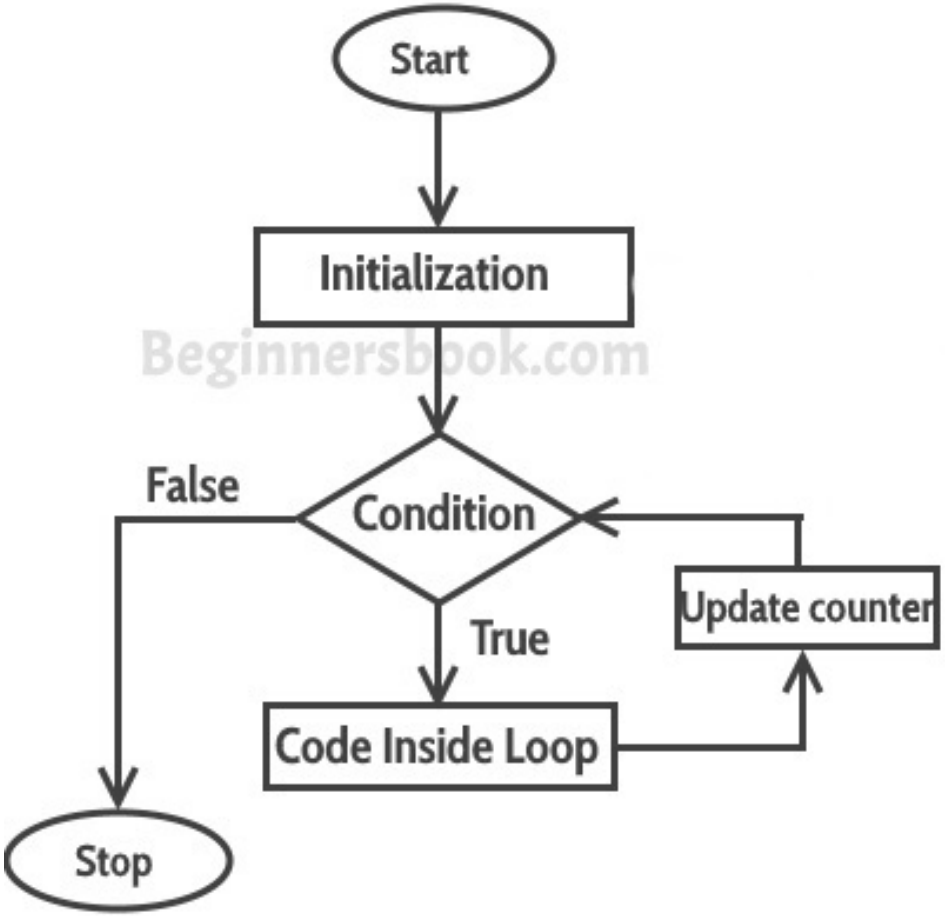

Figure: Flowchart for Loop structure

++ : increment operator

- The ++ operator is used make an increment of the number-based variables by 1
- increment operator (++) can be used in two ways
- When the increment operator is placed to the left of the variable, it will increment the variable's contents by 1 first, before it is used in another expression.

Example

```
main()
{
int i;
for (i = 1; i <= 3; i++)
{
printf("%d\n", i);
}
}
```

$--$ decrement operator; and $+=$ assignment operator

- The decrement operator is used in two ways to demonstrate how number based variables can be decremented by 1.
- $+=$ operator increments a variable to a new value plus itself

Example

- $x = y$ means - allocate the data in the y variable to the x variable, so $+$ here is an assignment operator
- $+=$ operator is also considered an assignment operator

Flow chart: Factorial of a number *n*

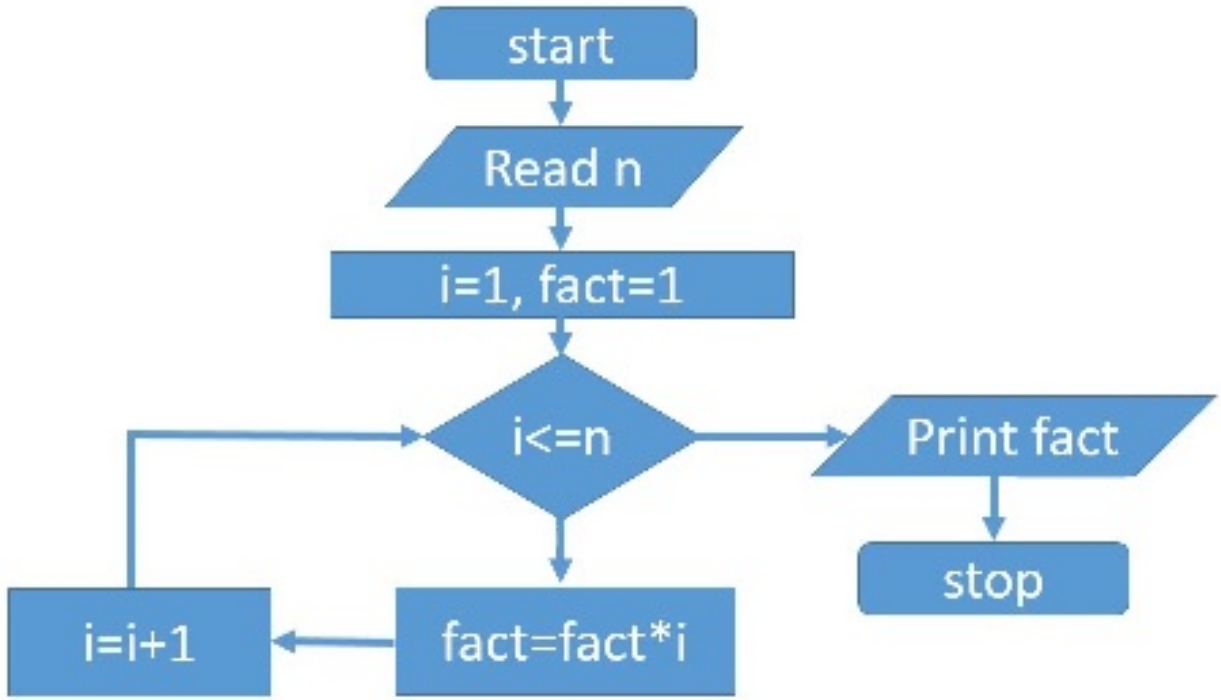

Figure: Flowchart Factorial

```
main() {
int n, i;
unsigned long long fact = 1;
printf("Enter an integer:  ");
 scanf("%d", &n);
 if (n < 0)
 printf("Error!  Factorial of a negative number doesn't
exist.");
 else {
for (i = 1; i <= n; ++i) {
fact *= i;
}
printf("Factorial of %d = %llu\n", n, fact);
}
}
```