Structured Program Development

- Thorough understanding of the problem is needed.
- Think of a careful approach to solve a problem
- Devise a structured sequence of actions.

Pseudocode

- Artificial and informal language
- A great help in design of an Algorithm
- It can be converted into “C” code
- It consists purely of characters
- It consist of action statements only.
- An informal program development.

Algorithms

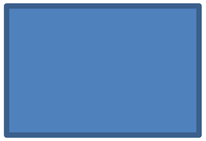
- Series of actions to be executed.
- The order in which these actions are to be executed must be right.
- Wrong order of execution causes chaos.
- Results in unexpected outcomes.

Algorithm Representation

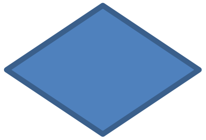
- Flow chart provides a graphical representation of an algorithm.
- Flow chart uses special purpose symbols.
- Rectangle, Diamond, Ovals, small circles are the symbols used in a flow chart.
- Symbols are connected by arrows called flow-lines.

Meaning of symbols

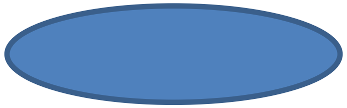
- Rectangle. (Action Symbol)



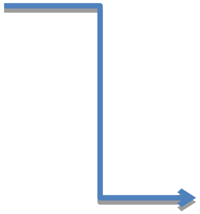
- Diamond. (Decision Symbol)



- Oval (Begin and end, or input/output)



- Flow lines



Control Structure

- Statements are executed sequentially
- Transfer of Control.
- This means statements that enable the programmer to choose the next line of code to be executed.
- Structured programming means elimination of the “goto” statement.

Control Structures…….cont…

There are three control structures:

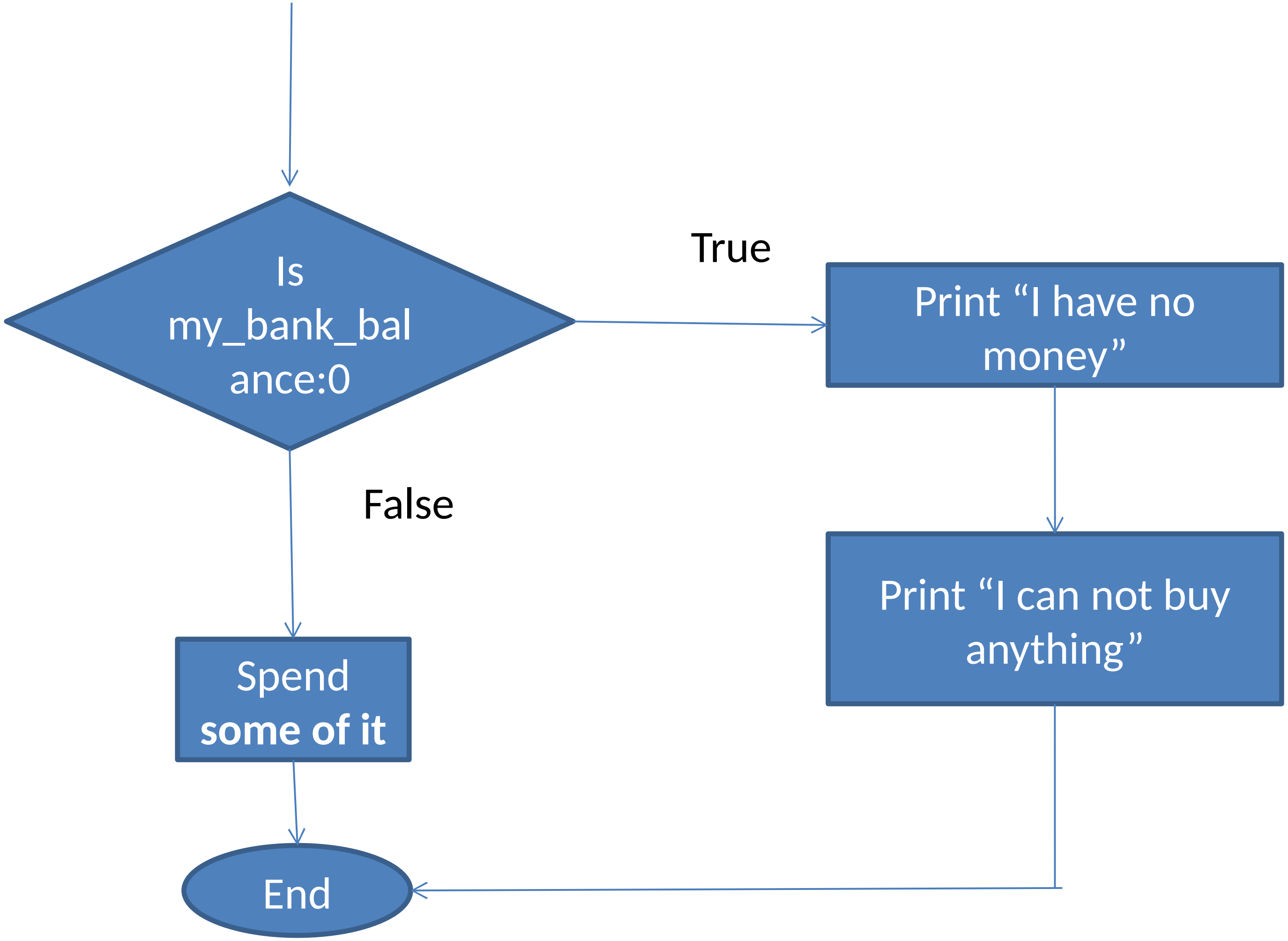
- **Sequence structure.** (Built into "C") each statement gets executed one after the other.
- **Selection Structure**. There are selection decision, made in order to address the next statement for execution.
- **Repetition structure.** Creation of loops for executing lines of codes again and again.

Selection Structure

“C” provides three types of selection structure

- The if selection structure. Select an action if a condition is true.
- The if/else. Selects to perform an action if a condition is true and performs a different action if the condition is false
- The switch selection structure. Performs one of many different action depending on the value of an expression.

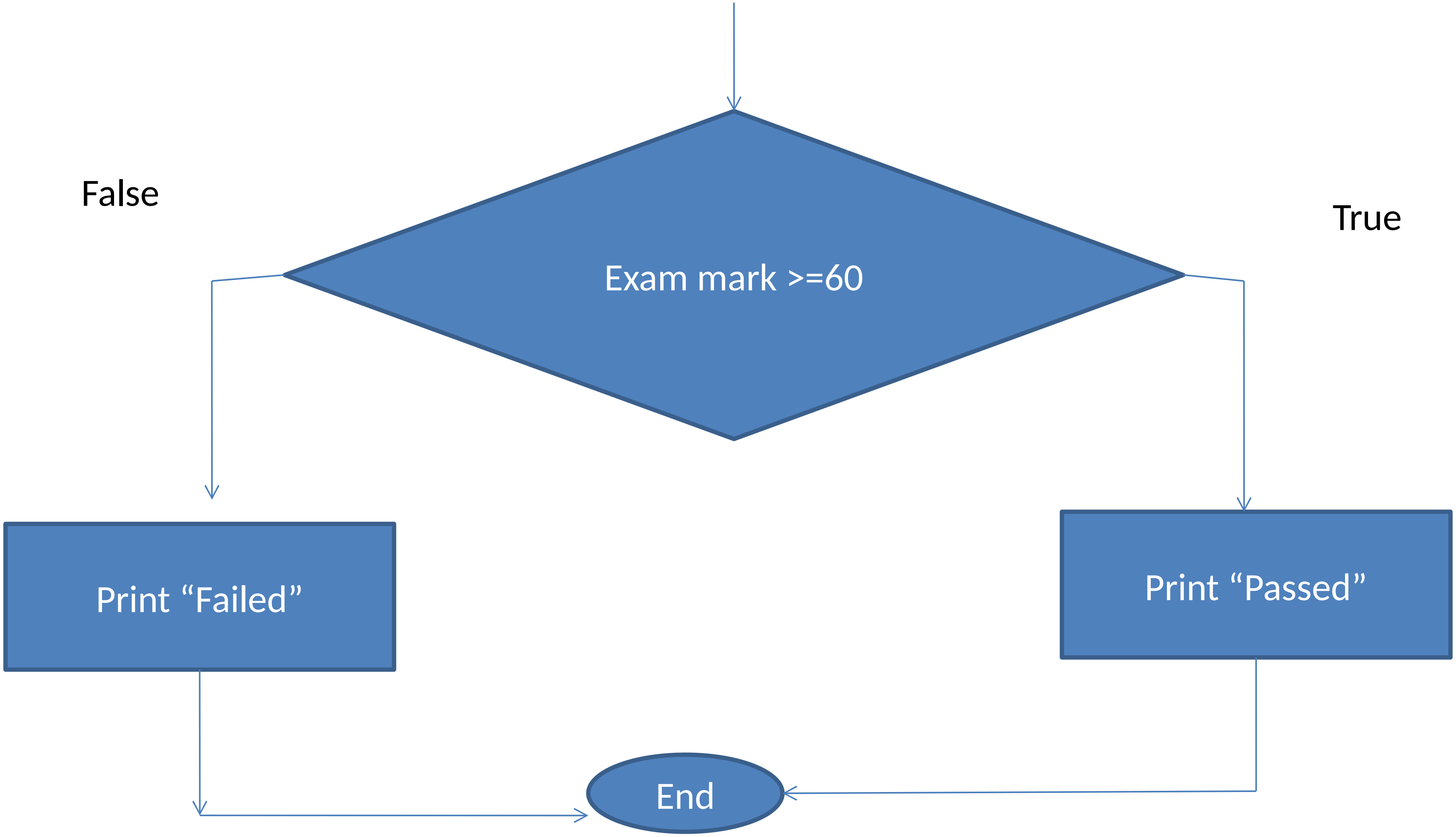
Flowchart for “if "statement



The “if” statement

```
If ( my_bank_balance==0)
        {
      Serialprintln(“I have no money”);
       Serialprintln(“ I can not buy anything);
          }
```

Flow chart for “if-else” statement



“If-else” statement

```
If (exam_mark >= 60)
    serialprintln(“pass”);
    else
    serialprintln(“failed”);
```

Nested “if/else”

Pseudo representation:

```
If student’s grade is grater than or equal to 70
    Print “A”
else
     if student’s grade is grater than or equal to 60
         Print “B”
     else
    if student’s grade is grater than or equal to 50
        Print “C”
              else
             If student’s grade grater than or equal to 40
                 Print “D”
    else
          Print “F”
```

“C”, Coded Nested “if/else” statement

```
If (grade >= 70)
  Serialprintln(“ A”);
else
  If (grade >=60)
   Serialprintln(“ B”);
  else
     If (grade >=50)
      Serialprintln(“ C”);
     else
        If (grade >=40)
               Serialprintln(“ D”);
        else
           Serialprintln(“ F”);
```

Different style of writing “C” Code

```
If (grade >= 70)    Serialprintln(“ A”);
else
   If (grade >=60) Serialprintln(“ B”);
   else
      If (grade >=50) Serialprintln(“ C”);
      else
         If (grade >=40) Serialprintln(“ D”);

         else Serialprintln(“ F”);
```