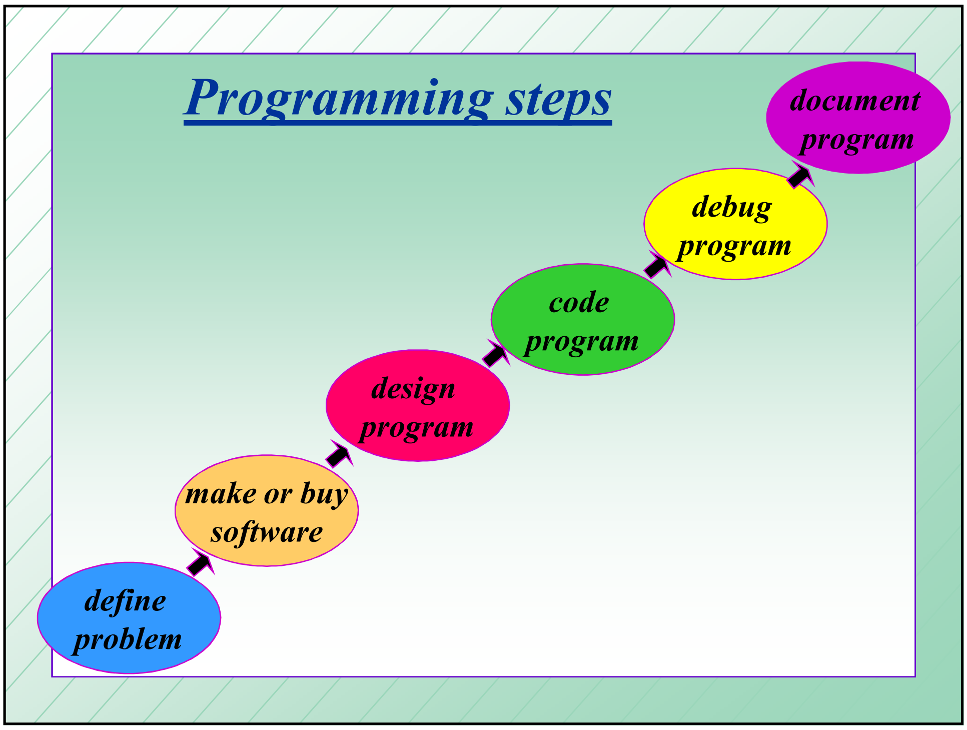
Lodz University of Technology
Faculty of Electrical, Electronic, Computer and Control Engineering

Introduction to algorithmics

Prof. Lidia Jackowska-Strumiłło
Dr Anna Fabijańska

Programming steps


define problem

make or buy software

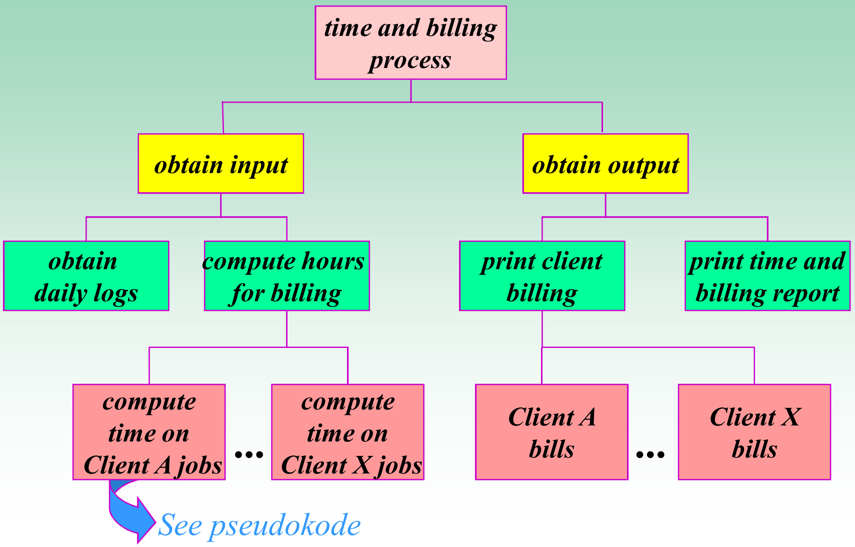
design program

code program

debug program

document program

Top-down program design

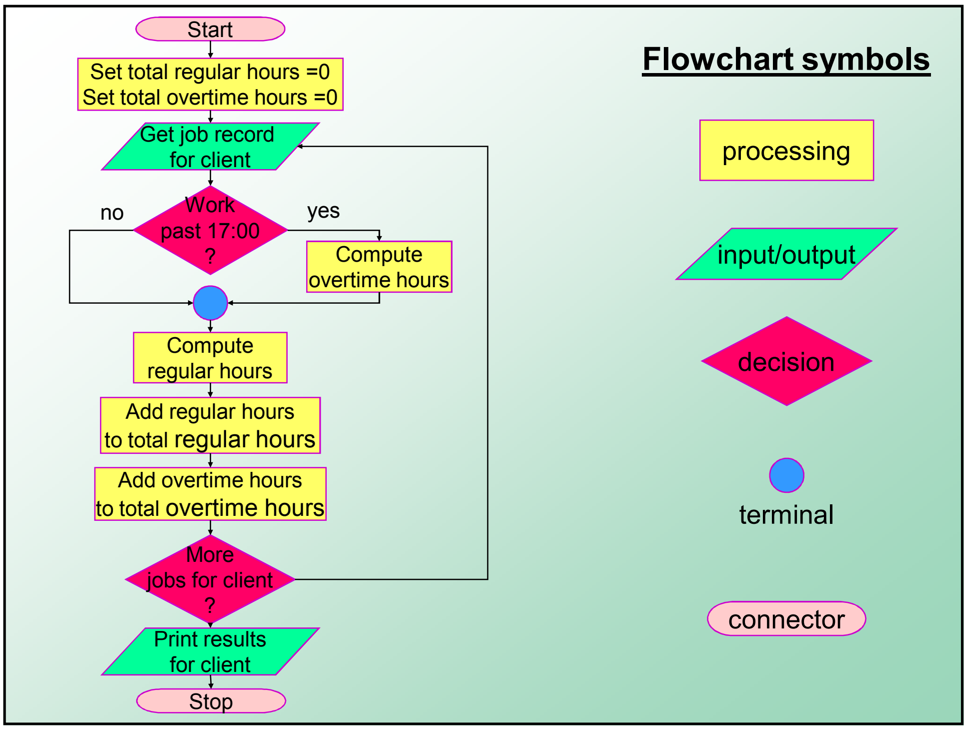


See pseudokode

Example of pseudocode

Compute time for Client A

- Set total regular hours and total overtime hours to zero.
- Get time in and time out for a job.
- If worked past 17:00 hours, then compute overtime hours.
- Compute regular hours.
- Add regular hours to total regular hours.
- Add overtime hours to total overtime hours.
- If there are more jobs for that client, go back and compute for that job as well.



Start

Set total regular hours =0
Set total overtime hours =0

Get job record for client

Work past 17:00 ?

no

yes

Compute overtime hours

Compute regular hours

Add regular hours to total regular hours

Add overtime hours to total overtime hours

More jobs for client ?

Print results for client

Stop

Flowchart symbols

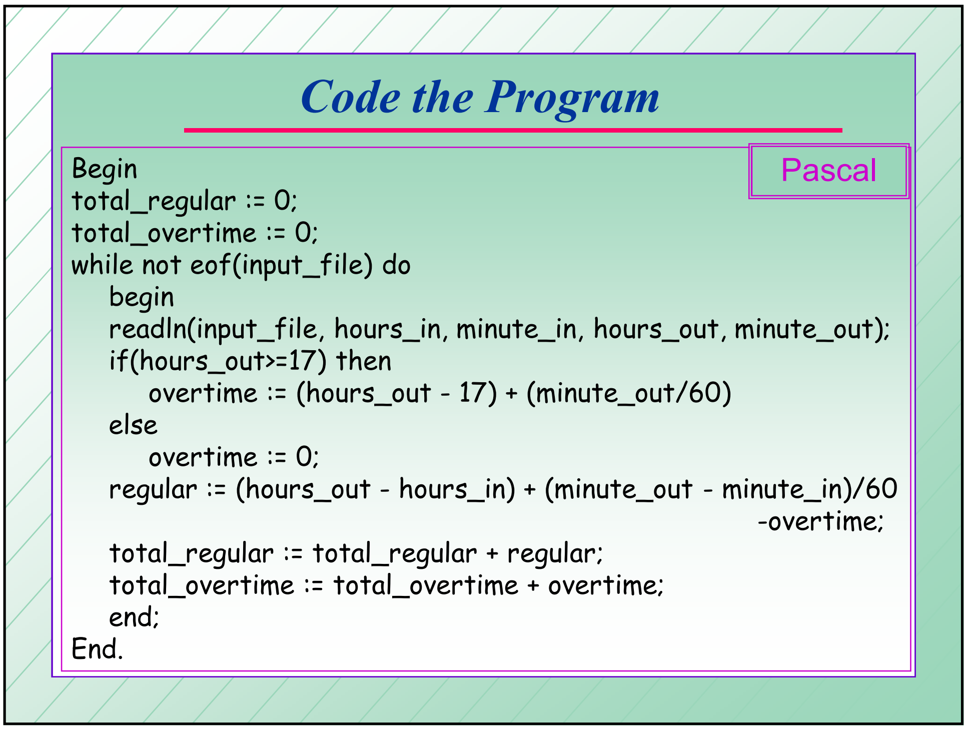
processing

input/output

decision

terminal

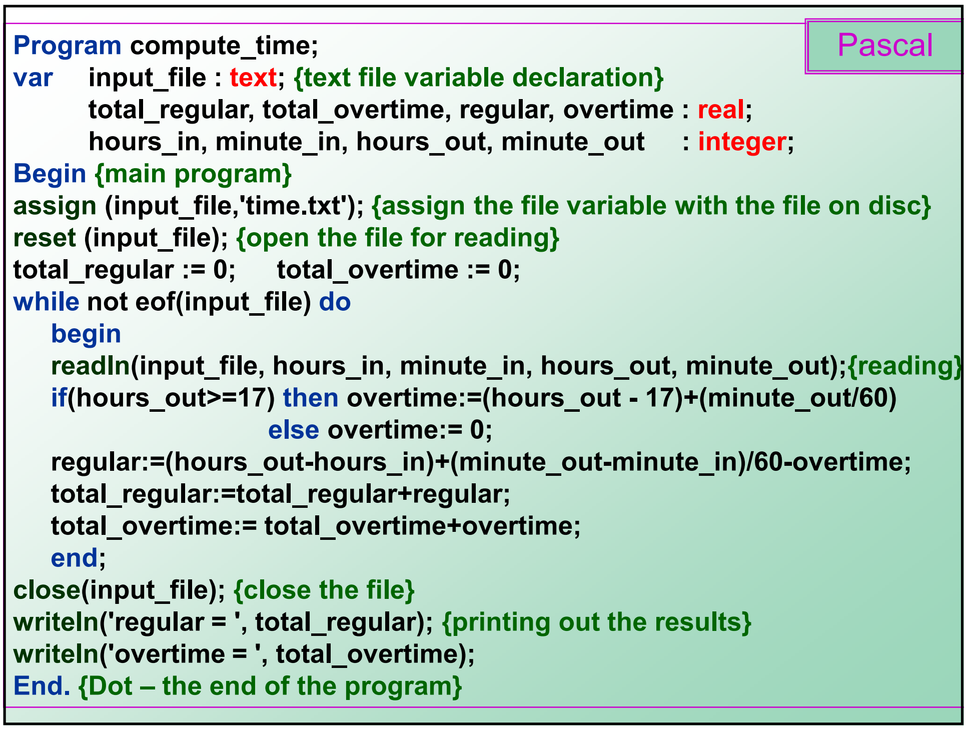
connector



Code the Program

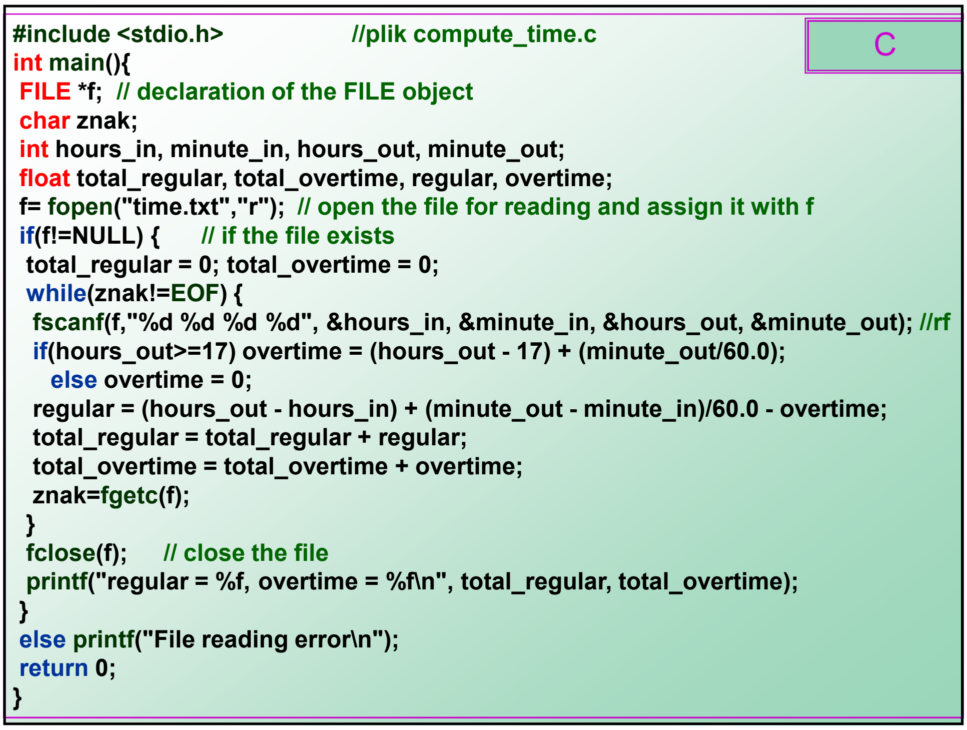
Pascal

```
Begin
total_regular := 0;
total_overtime := 0;
while not eof(input_file) do
    begin
    readln(input_file, hours_in, minute_in, hours_out, minute_out);
    if(hours_out>=17) then
        overtime := (hours_out - 17) + (minute_out/60)
    else
        overtime := 0;
    regular := (hours_out - hours_in) + (minute_out - minute_in)/60
                                                          -overtime;
    total_regular := total_regular + regular;
    total_overtime := total_overtime + overtime;
    end;
End.
```



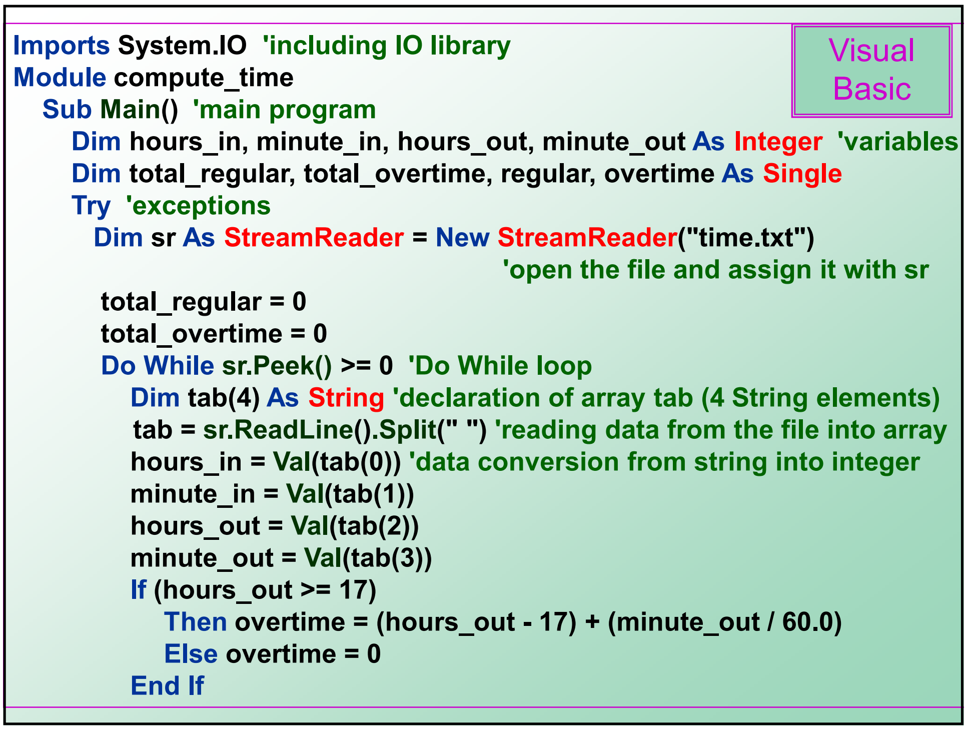
Pascal

```
Program compute_time;
var    input_file : text; {text file variable declaration}
       total_regular, total_overtime, regular, overtime : real;
       hours_in, minute_in, hours_out, minute_out     : integer;
Begin {main program}
assign (input_file,'time.txt'); {assign the file variable with the file on disc}
reset (input_file); {open the file for reading}
total_regular := 0;     total_overtime := 0;
while not eof(input_file) do
    begin
    readln(input_file, hours_in, minute_in, hours_out, minute_out);{reading}
    if(hours_out>=17) then overtime:=(hours_out - 17)+(minute_out/60)
                           else overtime:= 0;
    regular:=(hours_out-hours_in)+(minute_out-minute_in)/60-overtime;
    total_regular:=total_regular+regular;
    total_overtime:= total_overtime+overtime;
    end;
close(input_file); {close the file}
writeln('regular = ', total_regular); {printing out the results}
writeln('overtime = ', total_overtime);
End. {Dot – the end of the program}
```



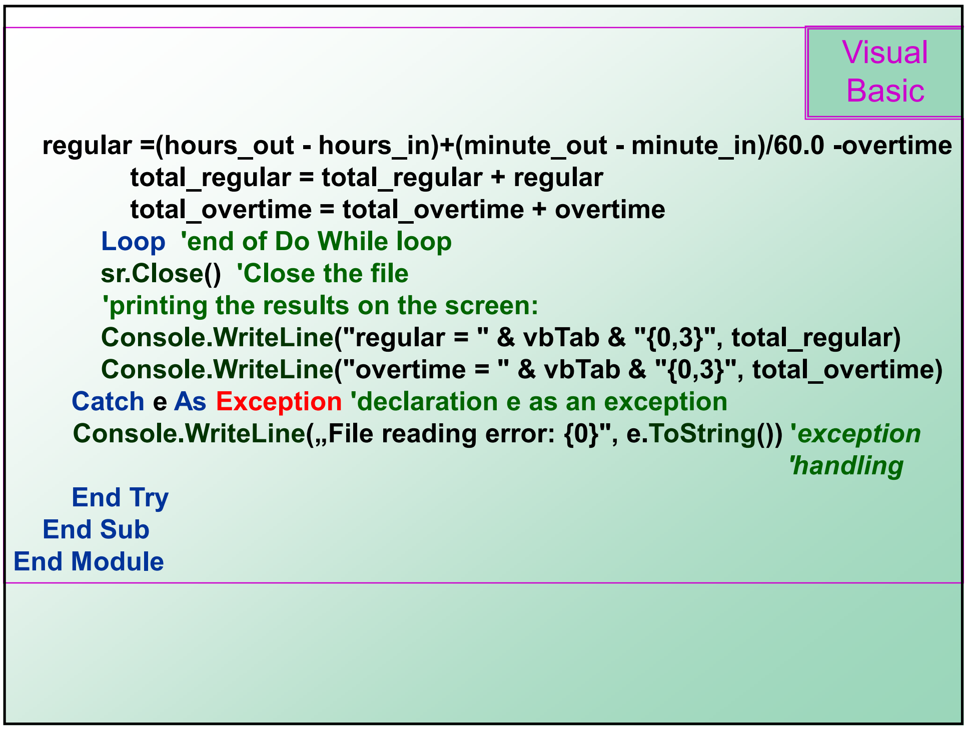
C

```
#include <stdio.h>                   //plik compute_time.c
int main(){
 FILE *f;  // declaration of the FILE object
 char znak;
 int hours_in, minute_in, hours_out, minute_out;
 float total_regular, total_overtime, regular, overtime;
 f= fopen("time.txt","r");  // open the file for reading and assign it with f
 if(f!=NULL) {      // if the file exists
  total_regular = 0; total_overtime = 0;
  while(znak!=EOF) {
   fscanf(f,"%d %d %d %d", &hours_in, &minute_in, &hours_out, &minute_out); //rf
   if(hours_out>=17) overtime = (hours_out - 17) + (minute_out/60.0);
     else overtime = 0;
   regular = (hours_out - hours_in) + (minute_out - minute_in)/60.0 - overtime;
   total_regular = total_regular + regular;
   total_overtime = total_overtime + overtime;
   znak=fgetc(f);
  }
  fclose(f);     // close the file
  printf("regular = %f, overtime = %f\n", total_regular, total_overtime);
 }
 else printf("File reading error\n");
 return 0;
}
```



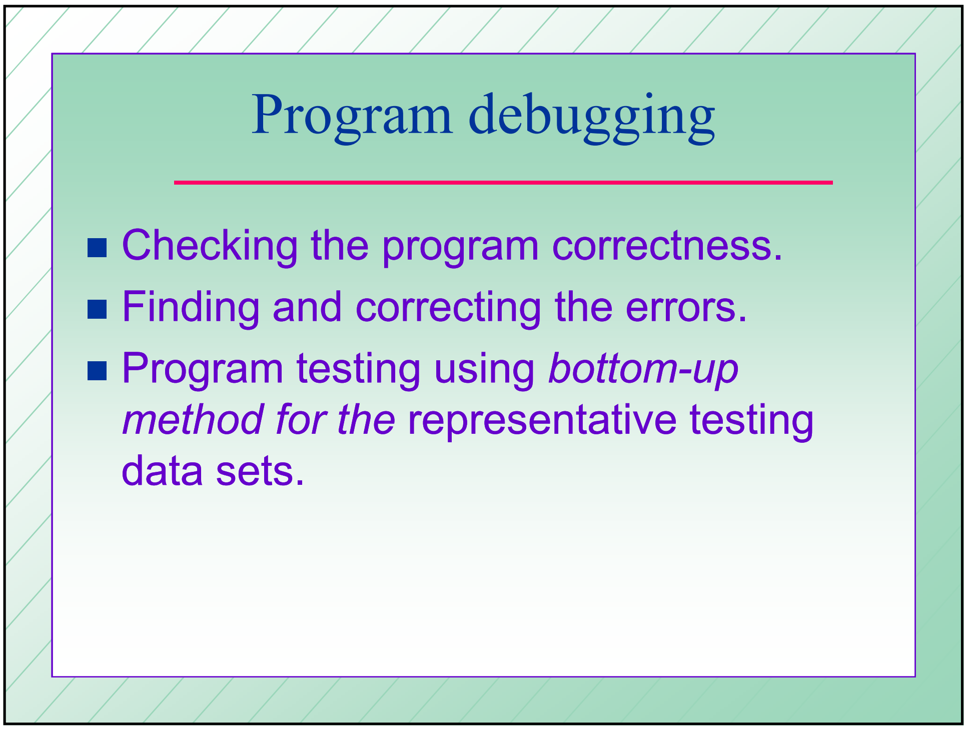
Visual Basic

```
Imports System.IO  'including IO library
Module compute_time
   Sub Main()  'main program
      Dim hours_in, minute_in, hours_out, minute_out As Integer  'variables
      Dim total_regular, total_overtime, regular, overtime As Single
      Try  'exceptions
        Dim sr As StreamReader = New StreamReader("time.txt")
                                             'open the file and assign it with sr
         total_regular = 0
         total_overtime = 0
         Do While sr.Peek() >= 0  'Do While loop
            Dim tab(4) As String 'declaration of array tab (4 String elements)
            tab = sr.ReadLine().Split(" ") 'reading data from the file into array
            hours_in = Val(tab(0)) 'data conversion from string into integer
            minute_in = Val(tab(1))
            hours_out = Val(tab(2))
            minute_out = Val(tab(3))
            If (hours_out >= 17)
               Then overtime = (hours_out - 17) + (minute_out / 60.0)
               Else overtime = 0
            End If
```



Visual Basic

```
   regular =(hours_out - hours_in)+(minute_out - minute_in)/60.0 -overtime
            total_regular = total_regular + regular
            total_overtime = total_overtime + overtime
         Loop  'end of Do While loop
         sr.Close()  'Close the file
         'printing the results on the screen:
         Console.WriteLine("regular = " & vbTab & "{0,3}", total_regular)
         Console.WriteLine("overtime = " & vbTab & "{0,3}", total_overtime)
      Catch e As Exception 'declaration e as an exception
      Console.WriteLine(„File reading error: {0}", e.ToString()) 'exception
                                                                 'handling
      End Try
   End Sub
End Module
```



Program debugging

- Checking the program correctness.
- Finding and correcting the errors.
- Program testing using *bottom-up method for the* representative testing data sets.