EECS 10: Computational Methods in Electrical and Computer Engineering

Lecture 5

Rainer Dömer

doemer@uci.edu

The Henry Samueli School of Engineering
Electrical Engineering and Computer Science
University of California, Irvine

Lecture 5.1: Overview

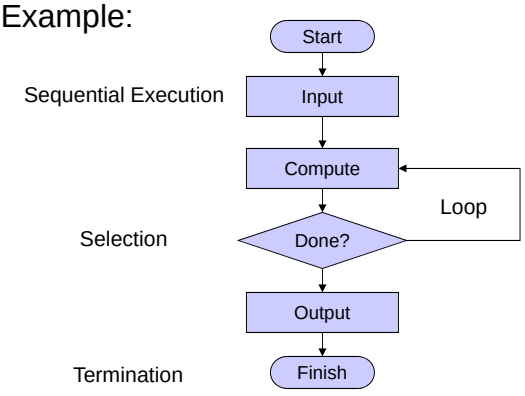
- Think before you program!
- Structured Programming
  - Control flow charts
  - Sequential statements
  - Conditional statements
  - Repetition statements
    - **`while`** loop
    - **`do-while`** loop
    - **`for`** loop
- Program Development
  - Example **`Interest.c`**

Programming == Thinking

- Programming ...
  - ... is *not* a mechanic procedure!
  - ... requires *thinking*!
- Program ...
  - ... *writing* requires an *intelligent human being*!
  - ... *execution* can be performed by a *dumb machine*.
- General programming steps:
  1. Understand the problem
  2. Define the input and output data
  3. Develop the algorithm (and specify it in pseudo code)
  4. Define the control flow (e.g. use control flow charts)
  5. Write the program in programming language
  6. Compile, test and debug the program

Structured Programming

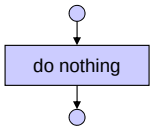
- Control flow charts
  - Graphical representation of program control flow
  - Example:



Structured Programming

- Empty statement blocks
  - empty compound statement
  - does nothing (no operation, no-op)
  - Example: Flow chart:

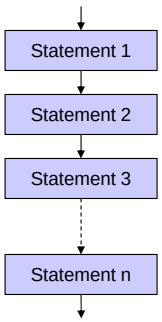
```
{
 /* nothing */
}
```



Structured Programming

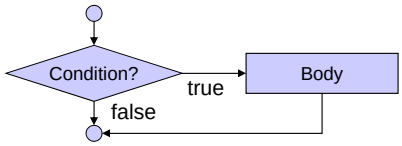
- Sequential execution in C
  - Statement blocks: *Compound statements*
  - Sequence of statements grouped by braces: **{ }**
- Example: Flow chart:

```
{
  /* statement 1 */

  /* statement 2 */

  /* statement 3 */

  /* ... */

  /* statement n */
}
```



Structured Programming

- Selection: **`if`** statement
  - Flow chart:

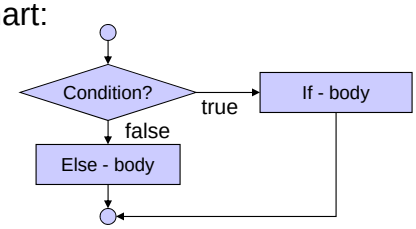


  - Example:

```
if (grade >= 60)
   { printf("You passed.");
    } /* fi */
```


Structured Programming

- Selection: **`if-else`** statement
  - Flow chart:

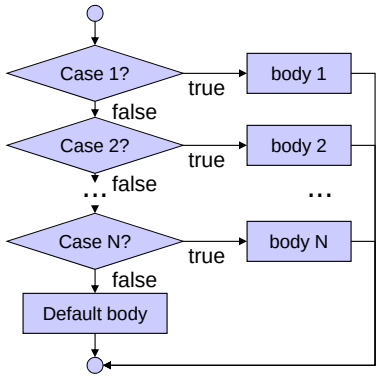


  - Example:

```
if (grade >= 60)
   { printf("You passed.");
    } /* fi */
else
   { printf("You failed.");
    } /* esle */
```

Structured Programming

- Selection: **`switch`** statement
  - Flow chart:

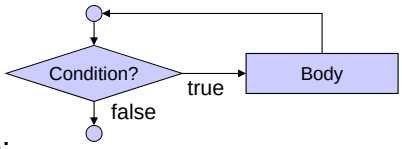


Example:

```
switch(LetterGrade)
{ case 'A':
   { printf("Excellent!");
     break; }
  case 'B':
  case 'C':
  case 'D':
   { printf("Passed.");
     break; }
  case 'F':
   { printf("Failed!");
     break; }
  default:
   { printf("Invalid grade!");
     break; }
} /* hctiws */
```


Structured Programming

- Repetition: **`while`** loop
  - Flow chart:



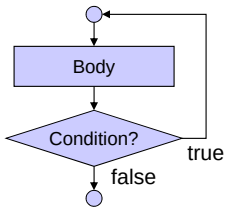
  - Example:

```
int product = 2;
while (product < 1000)
   { product *= 2;
    } /* elihw */
```

  - Note:
    - The condition is evaluated at the *beginning* of each loop!

Structured Programming

- Repetition: **`do-while`** loop
  - Flow chart:



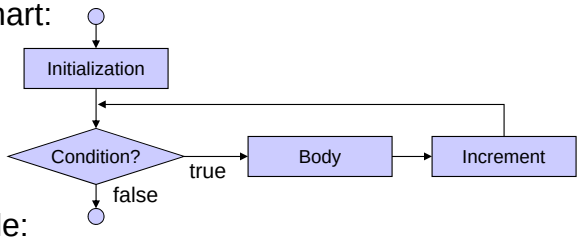
  - Example:

```
int product = 2;
do { product *= 2;
    } while (product < 1000);
```

  - Note:
    - The condition is evaluated at the *end* of each loop!

Structured Programming

- Repetition: **`for`** loop
  - Flow chart:



  - Example:

```
for(i = 0; i < 10; i++)
   { printf("i = %d\n", i);
    } /* rof */
```

  - Syntax:
    - **`for(`*`initialization`*`; `*`condition`*`; `*`increment`*`)`**
      `{ `*`body`*` }`

Program Development Example

- Compound interest: **`Interest.c`**
- Assignment:
  - Write a program that calculates the interest accumulated in a savings account. Given an initial deposit amount and an annual percentage rate (APR), compute the yearly interest earned and the resulting balance, for a period of ten years.
  - For example, for $1000 in savings at 4.5% APR, the annual interest should be tabulated as follows:

```
Interest for year  1 is $   45.00, total balance is $ 1045.00.
Interest for year  2 is $   47.02, total balance is $ 1092.03.
Interest for year  3 is $   49.14, total balance is $ 1141.17.
...
```


Program Development Example

- Compound interest: **`Interest.c`**
- Assignment:
  - Write a program that calculates the interest accumulated in a savings account. Given an initial deposit amount and an annual percentage rate (APR), compute the yearly interest earned and the resulting balance, for a period of ten years.

➢ Step 1: Understand the problem

- What is given?
  - deposit amount, annual percentage rate
- What is asked for?
  - yearly interest, resulting balance
- How do we compute what is asked for?
  - *interest = amount * APR/100*
  - *balance = amount + interest*

Program Development Example

- Step 1: Understand the problem
  - What is given?
    - deposit amount, annual percentage rate
  - What is asked for?
    - yearly interest, resulting balance

➢ Step 2: Define the input and output data
  - Input:
    - Deposit amount: **`amount`**, floating point type
    - Annual percentage rate: **`rate`**, floating point type
  - Output:
    - Current year: **`year`**, integral type
    - Interest earned: **`interest`**, floating point type
    - Resulting balance: **`balance`**, floating point type

Program Development Example

- Step 2: Define the input and output data
  - Deposit amount: **`amount`**, floating point type
  - Annual percentage rate: **`rate`**, floating point type
  - Current year: **`year`**, integral type
  - Interest earned: **`interest`**, floating point type
  - Resulting balance: **`balance`**, floating point type

➢ Step 3: Develop the algorithm (in pseudo code)
  - First, input **`amount`** and **`rate`**
  - For the current **`year`**, compute **`interest`** on the **`amount`**
  - Next, compute new **`balance`** at the end of the year
  - Then, print **`year`**, **`interest`** and **`balance`** in tabular format
  - Finally, set the **`amount`** to the new **`balance`**
  - Repeat the previous 4 steps for 10 years
  - Done!

Program Development Example

- Step 3: Develop the algorithm (in pseudo code)
  - First, input **amount** and **rate**
  - For the current **year**, compute **interest** on the **amount**
  - Next, compute new **balance** at the end of the year
  - Then, print **year**, **interest** and **balance** in tabular format
  - Finally, set the **amount** to the new **balance**
  - Repeat the previous 4 steps for 10 years
- ➢ Step 4: Define the control flow
  - First, input **amount** and **rate**
  - Repeat for 10 years:
    - Compute **interest** on the **amount**
    - Compute new **balance** at the end of the year
    - Print **year**, **interest** and **balance** in tabular format
    - Set the **amount** to the new **balance**
  - Done!

Program Development Example

- Step 4: Define the control flow
  - First, input **amount** and **rate**
  - Repeat for 10 years:
    - Compute **interest** on the **amount**
    - Compute new **balance** at the end of the year
    - Print **year**, **interest** and **balance** in tabular format
    - Set the **amount** to the new **balance**
- ➢ Step 5: Write the program in programming language

```
double amount;      double rate;      int year;
double interest;    double balance;
printf("Please enter the initial amount in $: ");
scanf("%lf", &amount);
printf("Please enter the interest rate in %% : ");
scanf("%lf", &rate);
```

etc.

Example Program

- Compound interest: **`Interest.c`** (part 1/2)

```
/* Interest.c: compound interest on savings account     */
/* author: Rainer Doemer                                */
/* modifications:                                       */
/* 10/18/06 RD  distinguish amount and balance          */
/* 10/19/04 RD  initial version                         */

#include <stdio.h>

/* main function */

int main(void)
{
   /* variable definitions */
   double amount, balance, rate, interest;
   int    year;

   /* input section */
   printf("Please enter the initial amount in $: ");
   scanf("%lf", &amount);
   printf("Please enter the interest rate in %% : ");
   scanf("%lf", &rate);
...
```


Example Program

- Compound interest: **`Interest.c`** (part 2/2)

```
...

   /* computation and output section */
   for(year = 1; year <= 10; year++)
      { interest = amount * (rate/100.0);
        balance = amount + interest;
        printf("Interest for year %2d is $%8.2f,"
               " total balance is $%8.2f.\n",
               year, interest, balance);
        amount = balance;
       } /* rof */

   /* exit */
   return 0;
} /* end of main */

/* EOF */
```

Program Development Example

- Step 5: Write the program in programming language
- ➢ Step 6: Compile, test (and debug) the program

```
% vi Interest.c
% gcc Interest.c -o Interest -Wall -ansi
% Interest
Please enter the initial amount in $: 1500
Please enter the interest rate in % : 1.5
Interest for year  1 is $   22.50, total balance is $ 1522.50.
Interest for year  2 is $   22.84, total balance is $ 1545.34.
Interest for year  3 is $   23.18, total balance is $ 1568.52.
Interest for year  4 is $   23.53, total balance is $ 1592.05.
Interest for year  5 is $   23.88, total balance is $ 1615.93.
Interest for year  6 is $   24.24, total balance is $ 1640.16.
Interest for year  7 is $   24.60, total balance is $ 1664.77.
Interest for year  8 is $   24.97, total balance is $ 1689.74.
Interest for year  9 is $   25.35, total balance is $ 1715.08.
Interest for year 10 is $   25.73, total balance is $ 1740.81.
%
```