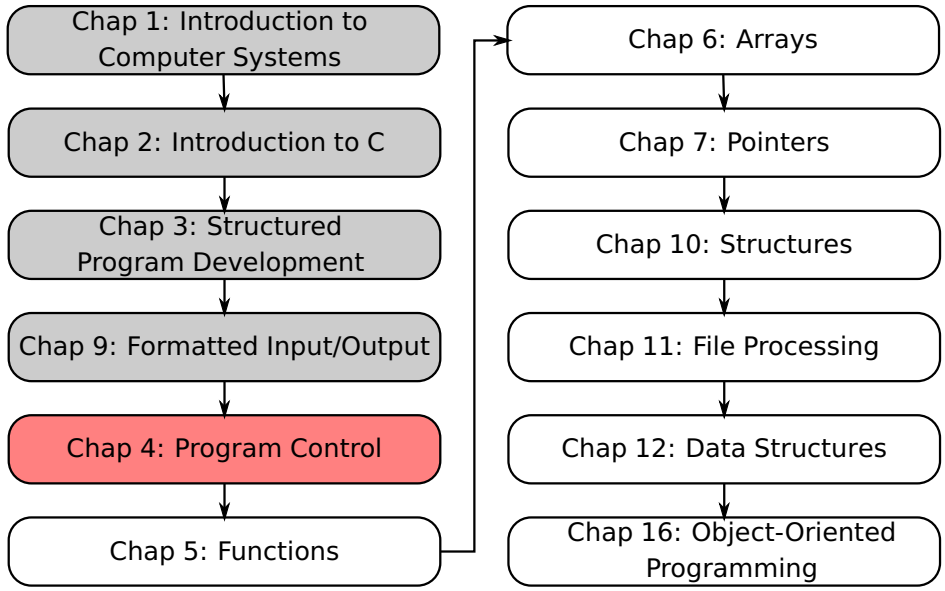
Computer Programming 143 – Lecture 9
Program Flow Control III

Electrical and Electronic Engineering Department
University of Stellenbosch

Corné van Daalen
Thinus Booysen
Willie Smit
Willem Jordaan

Module Overview



Lecture Overview

4.7 The `switch` Multiple-Selection Statement I

Unelegant `if` statement

```
if ( condition ) {
    statement(s);
}
else if ( condition ) {
    statement(s);
}
else if ( condition ) {
    statement(s);
}
else if ( condition ) {
    statement(s);
}
...
```

4.7 The `switch` Multiple-Selection Statement II

`switch`

- Useful when a variable or expression is tested for all the values it can assume and different actions are taken

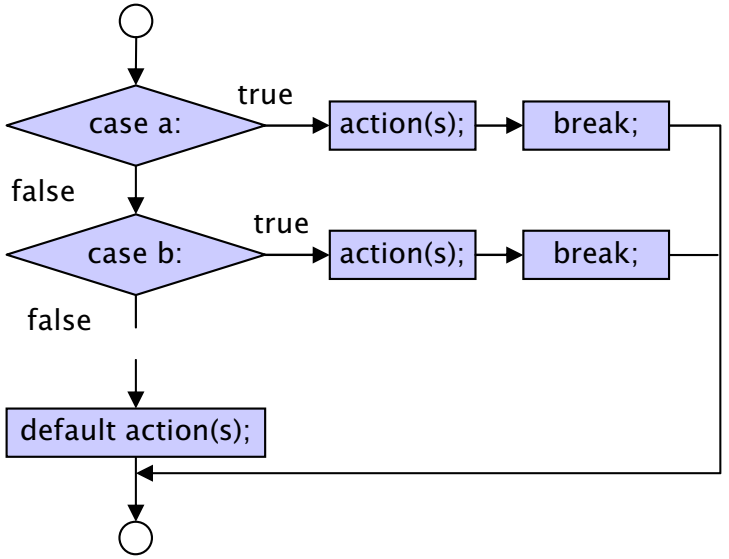
Format:

- Series of **`case`** labels and an optional **`default`** case

```
switch ( value ) {
    case 1 : action(s);
    break;
    case 2 : action(s);
    break;
    default: action(s);
}
```

- **`break`**; exits from statement

4.7 The `switch` Multiple-Selection Statement III



4.7 `switch` Program Example I

Example C code

```
/* Counting Grades (based on Fig. 4.7 in Deitel & Deitel) */
#include <stdio.h>
#include <stdlib.h>
// function main begins program execution
int main( void )
{
  int numStudents; //total number of students
  char grade; // grade input variable
  int passCount = 0; // number of passes
  int failCount = 0; // number of failures

  setbuf(stdout, 0);

  printf("Enter the total number of students.\n");
  scanf("%d",&numStudents); //read the total number of students
  printf( "Enter results (P for pass; F for failure).\n");
  do {
     scanf(" %c", &grade);//scanf reads one charater from keyboard
```


4.7 `switch` Program Example II

Example C code (cont'd...)

```
switch ( grade ) { // switch nested in while
   case 'F':       // grade input was uppercase F
   case 'f':       // or lowercase f
      failCount++; // increment number of failures
      break;       // exit switch

   case 'P':       // grade input was uppercase P
   case 'p':       // or lowercase p
      passCount++; // increment number of passes
      break;       // exit switch

   case '\n':      // ignore newlines
   case '\t':      // tabs,
   case ' ':       // and spaces in input
      break;       // exit switch
```

4.7 switch Program Example III

Example C code (cont'd...)

```
        default:        // catch all other characters
           printf( "Invalid character entered. " );
           printf( "Enter a new result (P or F).\n" );
           break;
     } // end switch
   } while ((passCount+failCount) < numStudents);// end do....while

   // output summary of results
   printf( "\nTest results:\n" );
   printf( "Passed: %d\n", passCount ); // display number of passes
   printf( "Failed: %d\n", failCount ); // display number of failures

   return 0; // indicate program ended successfully
} // end function main
```


4.7 switch Program Example IV

Example output

```
Enter the total number of students.
5
Enter results (P for pass; F for failure).
p
P
f
F
k
Invalid character entered. Enter a new result (P or F).
P

Test results:
Passed: 3
Failed: 2
```


4.9 The break Statement I

break

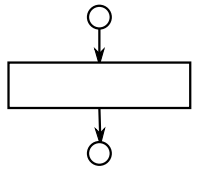
- Causes immediate exit from a **while**, **for**, **do...while** or **switch** statement
- Program execution continues with the first statement after the structure
- Common uses of the **break** statement
  - Escape early from a loop
  - Skip the remainder of a **switch** statement

Remarks

- Violates rules of structured programming when used with **while**, **for** or **do...while** statement
- For more information refer to the textbook

4.12 Summary of Structured Programming I

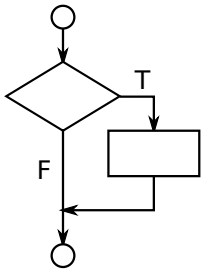
Sequence structure



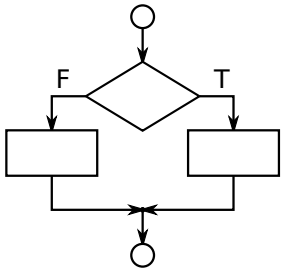
4.12 Summary of Structured Programming II

Selection structures

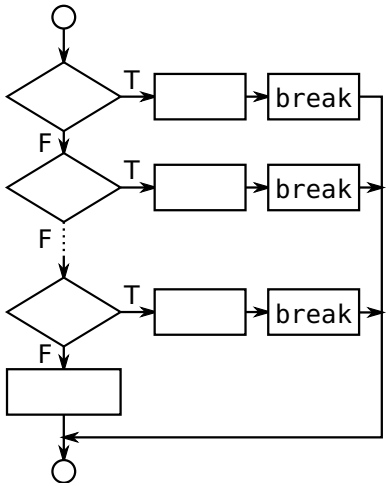
`if` statement



`if...else` statement



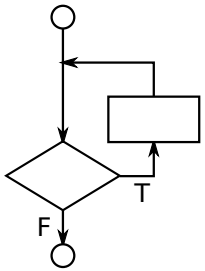
`switch` statement



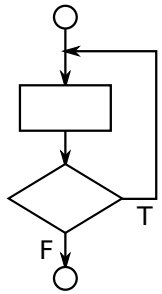
4.12 Summary of Structured Programming III

Repetition structures

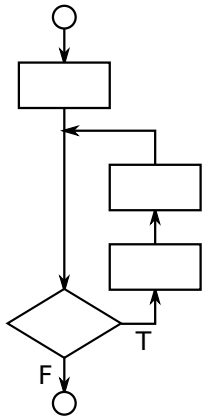
`while` statement



`do...while` statement

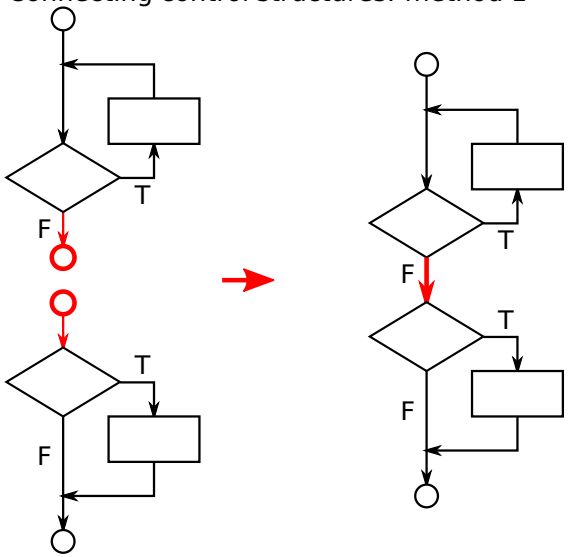


`for` statement



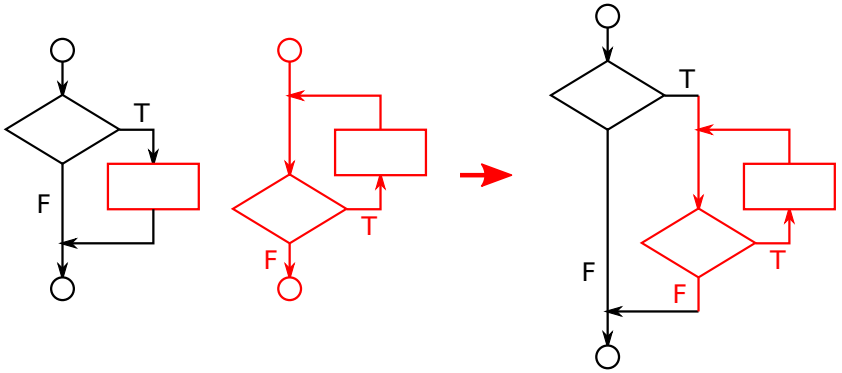
4.12 Summary of Structured Programming IV

Connecting control structures: method 1



4.12 Summary of Structured Programming V

Connecting control structures: method 2



Perspective

Today

Program flow control III

- `switch` selection structure
- `break` and `continue`
- Summary of structured program development

Next lecture

Functions I

- Introduction to functions

Homework

1. Study Sections 4.7, 4.9, 4.12 in Deitel & Deitel
2. Do Self Review Exercises 4.1, 4.2(a)&(b), 4.4(c) in Deitel & Deitel
3. Do Exercises 4.5(b)&(c), 4.19, 4.24