{C}

PROGRAMMING

IT 112: Introduction to Programming

Dr. Manish Khare

Dr. Bakul Gohel

Lecture 10

Case Control Instruction

- In real life we are often faced with situations where we are required to make a choice between a number of alternatives rather than only one or two.

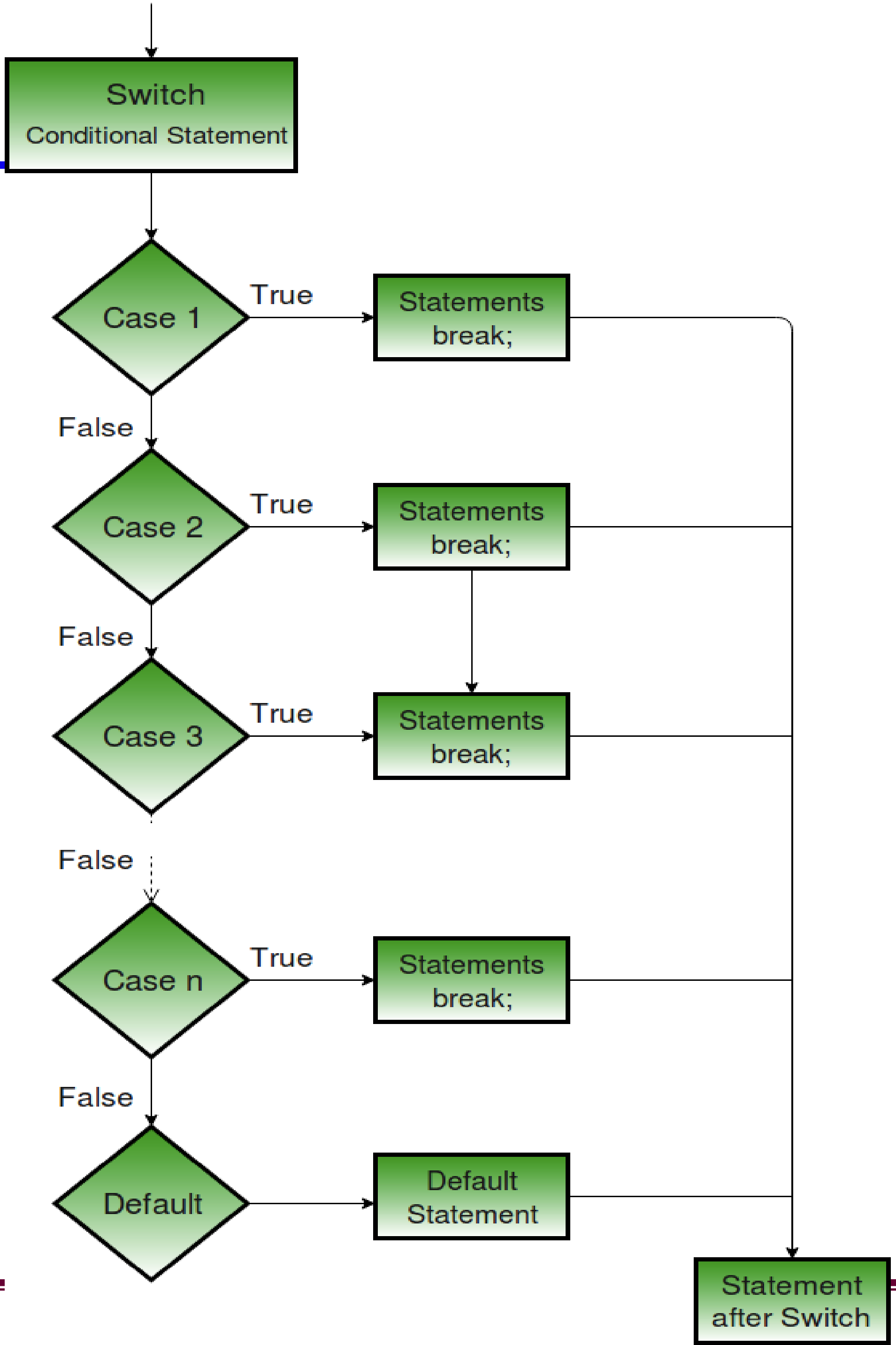
- C provides a special control statement that allows us to handle such cases effectively; rather than using a series of if statements.

Decision Using Switch

- The control statement that allows us to make a decision from the number of choices is called a **switch**, or more correctly a **switch-case-default**, since these three keywords go together to make up the control statement

```
switch ( integer expression )
 {
          case constant 1 :
                    do this ;
          case constant 2 :
                    do this ;
          case constant 3 :
                    do this ;
          default :
                    do this ;
 }
```

Decision Using Switch



- The integer expression following the keyword switch is any C expression that will yield an integer value.
- It could be an integer constant like 1, 2 or 3, or an expression that evaluates to an integer.
- The keyword case is followed by an integer or a character constant.
- Each constant in each case must be different from all the others.
- The “do this” lines in the above form of switch represent any valid C statement.

- What happens when we run a program containing a switch?
  - First, the integer expression following the keyword switch is evaluated.
  - The value it gives is then matched, one by one, against the constant values that follow the case statements.
  - When a match is found, the program executes the statements following that case, and all subsequent case and default statements as well.
  - If no match is found with any of the case statements, only the statements following the default are executed

```
main( )
{
int i = 2 ;
switch ( i )
{
case 1 :
    printf ( "I am in case 1 \n" ) ;
case 2 :
    printf ( "I am in case 2 \n" ) ;
case 3 :
    printf ( "I am in case 3 \n" ) ;
default :
    printf ( "I am in default \n" ) ;
}
}
```

The output of this program would be

```
I am in case 2
I am in case 3
I am in default
```

Surprised!!

- The output is definitely not what we expected!
- We didn’t expect the second and third line in the above output.
- The program prints case 2 and 3 and the default case. Well, yes.
- We said the switch executes the case where a match is found and all the subsequent cases and the default as well.

- If you want that only case 2 should get executed, it is upto you to get out of the switch then and there by using a break statement.

```
main( )
{
int i = 2 ;
switch ( i )
{
case 1 :
    printf ( "I am in case 1 \n" ) ;
    break ;
case 2 :
    printf ( "I am in case 2 \n" ) ;
    break ;
case 3 :
    printf ( "I am in case 3 \n" ) ;
    break ;
default :
    printf ( "I am in default \n" ) ; } }
```

The output of this program would be:

```
I am in case 2
```

- The previous example shows how this is done.

- Note that there is no need for a break statement after the default, since the control comes out of the switch anyway

Few tips about usage of switch

- The program in previous slides may give you an impression that cases in a switch must be arranged in ascending order – 1,2,3, and default.
- In fact, you can put the cases in any order

```
main( )
{
int i = 22 ;
switch ( i )
{
case 121 :
   printf ( "I am in case 121 \n" ) ;
   break ;
case 7 :
   printf ( "I am in case 7 \n" ) ;
   break ;
case 22 :
   printf ( "I am in case 22 \n" ) ;
   break ;
default :
   printf ( "I am in default \n" ) ; } }
```

The output of this program would be:

```
I am in case 22
```

- You are also allowed to use char values in case and switch as shown in the following program

```
main( )
{
char i = ‘x’ ;
switch ( i )
{
case ‘v’ :
    printf ( "I am in case v \n" ) ;
    break ;
case ‘a’ :
    printf ( "I am in case a \n" ) ;
    break ;
case ‘x’ :
    printf ( "I am in case x \n" ) ;
    break ;
default :
    printf ( "I am in default \n" ) ; } }
```

The output of this program would be:

```
I am in case x
```

In fact here when we use ‘v’, ‘a’, ‘x’ they are actually replaced by the ASCII values (118, 97, 120) of these character constants

- At times we may want to execute a common set of statements for multiple cases.
- How this can be done is shown in the following example

```
main( )
{
char ch ;
printf ( "Enter any of the alphabet a, b, or c " ) ;
scanf ( "%c", &ch ) ;
switch ( ch )
{ case 'a’ :
case 'A’ :
    printf ( "a as in ashar" ) ;
    break ;
case 'b’ :
case 'B’ :
    printf ( "b as in brain" ) ;
    break ;
case 'c’ :
case 'C’ :
    printf ( "c as in cookie" ) ;
    break ;
default :
     printf ( "wish you knew what are alphabets" ) ;
}
}
```

- Here, we are making use of the fact that once a case is satisfied the control simply falls through the case till it doesn't encounter a break statement.
- That is why if an alphabet a is entered the case 'a' is satisfied and since there are no statements to be executed in this case the control automatically reaches the next case i.e. case 'A' and executes all the statements in this case.

- Even if there are multiple statements to be executed in each case there is no need to enclose them within a pair of braces (unlike if, and else).

- Every statement in a switch must belong to some case or the other. If a statement doesn’t belong to any case the compiler won’t report an error. However, the statement would never get executed.

- If we have no default case, then the program simply falls through the entire switch and continues with the next instruction (if any,) that follows the closing brace of switch.

- The break statement when used in a switch takes the control outside the switch.
  - However, use of continue will not take the control to the beginning of switch as one is likely to believe.

- In principle, a switch may occur within another, but in practice it is rarely done. Such statements would be called nested switch statements.

- The switch statement is very useful while writing menu driven programs.

- We can check the value of any expression in a switch.
  - Thus the following switch statements are legal.
    - switch (i + j * k)
    - switch (23 + 45 % 4 * k)
    - switch (a < 4 && b > 7)
  - Expressions can also be used in cases provided they are constant expressions.
    - Thus case 3 + 7 is correct, however, case a + b is incorrect.

- Is switch a replacement for if? Yes or No
- Yes, because it offers a better way of writing programs as compared to if, and no because in certain situations we are left with no choice but to use if.
- The disadvantage of switch is that one cannot have a case in a switch which looks like:
  - case i <= 20 :
- The advantage of switch over if is that it leads to a more structured program and the level of indentation is manageable, more so if there are multiple statements within each case of a switch.

switch Versus if-else Ladder

- There are some things that you simply cannot do with a switch.
- These are:

A. A float expression cannot be tested using a switch

B. Cases can never have variable expressions (for example it is wrong to say case a +3 :)

C. Multiple cases cannot use same expressions.

- Thus the following switch is illegal:

```
switch ( a )
{
case 3 :
...
case 1 + 2 :
 ...
}
```

➢What would be the output

```
main( )
{
        char suite = 3 ;
        switch ( suite )
        {
        case 1 :
                printf ( "\nDiamond" ) ;
        case 2 :
                printf ( "\nSpade" ) ;
        default :
                printf ( "\nHeart") ;
        }
        printf ( "\nI thought one wears a suite" ) ;
}
```

➢What would be the output

```
main( )
{
        int k, j = 2 ;
        switch ( k = j + 1 )
        {
        case 0 :
                printf ( "\nTailor") ;
        case 1 :
                printf ( "\nTutor") ;
        case 2 :
                printf ( "\nTramp") ;
        default :
                printf ( "\nPure Simple Egghead!" ) ;
        }
}
```

➢What would be the output

```
main( )
{
        int k ;
        float j = 2.0 ;
        switch ( k = j + 1 )
        {
                case 3 :
                        printf ( "\nTrapped" ) ;
                        break ;
                default :
                        printf ( "\nCaught!" ) ;
        }
}
```

What would be the output

```
main( )
{
        int ch = 'a' + 'b’ ;
        switch ( ch )
        {
                case 'a’ :
                case 'b’ :
                        printf ( "\nYou entered b" ) ;
                case 'A’ :
                        printf ( "\na as in ashar" ) ;
                case 'b' + 'a’ :
                        printf ( "\nYou entered a and b" ) ;
        }
}
```

```
main( )
{
        int temp ;
        scanf ( "%d", &temp ) ;
        switch ( temp )
        {
                case ( temp <= 20 ) :
                        printf ( "\nOoooooohhhh! Damn cool!" ) ;
                case ( temp > 20 && temp <= 30 ) :
                        printf ( "\nRain rain here again!" ) ;
                case ( temp > 30 && temp <= 40 ) :
                        printf ( "\nWish I am on Everest" ) ;
                default :
                        printf ( "\nGood old nagpur weather" ) ;
        }
}
```

```
main( )
{
        float a = 3.5 ;
        switch ( a )
        {
                case 0.5 :
                        printf ( "\nThe art of C" ) ;
                break ;
                case 1.5 :
                        printf ( "\nThe spirit of C" ) ;
                break ;
                case 2.5 :
                        printf ( "\nSee through C" ) ;
                break ;
                case 3.5 :
                printf ( "\nSimply c" ) ;
        }
}
```

```
main( )
{
        int a = 3, b = 4, c ;
        c = b – a ;
        switch ( c )
        {
                case 1 || 2 :
                printf ( "God give me an opportunity to change
things" ) ;
                break ;

                case a || b :
                        printf ( "God give me an opportunity to
run my show" ) ;
                break ;
        }
}
```

- Write a menu driven program which has following options:
  - 1. Sum of all the numbers
  - 2. Prime or not
  - 3. Odd or even

Make use of switch statement.

https://ideone.com/CP6cYU

Exercise?

Write a program that takes name of animals (dog, cat, mouse) as input, and subsequently print sound they produce while barking.

- Create using if-else

- Create using switch-case-default

Home Assignment

- Write a program which to find the grace marks for a student using switch. The user should enter the class obtained by the student and the number of subjects he has failed in.
  - – If the student gets first class and the number of subjects he failed in is greater than 3, then he does not get any grace. If the number of subjects he failed in is less than or equal to 3 then the grace is of 5 marks per subject.
  - – If the student gets second class and the number of subjects he failed in is greater than 2, then he does not get any grace. If the number of subjects he failed in is less than or equal to 2 then the grace is of 4 marks per subject.
  - – If the student gets third class and the number of subjects he failed in is greater than 1, then he does not get any grace. If the number of subjects he failed in is equal to 1 then the grace is of 5 marks per subject

Solve using both if-else & switch-case-default