UEM
UNIVERSITY OF ENGINEERING & MANAGEMENT
Good Education, Good Jobs

Flow Control

Switch, Break, Continue, Goto, and Loops

THE *break* STATEMENT

- *break* allows us to jump out of a loop instantly, without waiting to get back to the conditional test.
- When *break* is encountered inside any loop, control automatically passes to the first statement after the loop.
- A break is usually associated with an *if*.

THE *break* STATEMENT

```
/*Check if a number is prime number*/

#include<stdio.h>

void main( )
{
   int num, i ;
   printf ( "Enter a number " ) ;
   scanf ( "%d", &num ) ;
   i = 2 ;
   while ( i <= num - 1 )
   {
      if ( num % i == 0 )
      {
```

THE *continue* STATEMENT

- The keyword *continue* allows us to take the control to the beginning of the loop, bypassing the statements inside the loop, which have not yet been executed.

```
/*Example to demonstrate continue*/

#include<stdio.h>

void main( )
{
   int i, j ;
   for ( i = 1 ; i <= 2 ; i++ )
   {
      for ( j = 1 ; j <= 2 ; j++ )
      {
         if ( i == j )
```

DECISIONS *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.
  - Case conditions can be arranged in any order.
  - Only int & char are allowed for case checking.
  - Common statements can run using multiple cases.
  - 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).

DECISIONS *USING switch*

- 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. For example, in the following program the printf( ) never goes to work.

```
main( )
{
        int i, j ;
        printf ( "Enter value of i" ) ;
        scanf ( "%d", &i ) ;
        switch ( i )
        {
                printf ( "Hello" ) ;
                case 1 :
                        j = 10 ;
                        break ;
                case 2 :
                        j = 20 ;
                        break ;
        }
}
```

THE *goto* KEYWORD

```
/*goto demo-2*/
#include<stdio.h>
void main( )
{
   int goals ;
   printf ( "Enter the number of goals scored against India " ) ;
   scanf ( "%d", &goals ) ;
   if ( goals <= 5 )
      goto sos ;
   else
   {
      printf ( "About time soccer players learnt C\n" ) ;
      printf ( "and said goodbye! adieu! to soccer" ) ;
      exit(0) ; /* terminates program execution */
   }
   sos :
      printf ( "To err is human!" ) ;
}
```

- If the condition is satisfied the goto statement transfers control to the label ‘sos’, causing printf( ) following sos to be executed.
- Any number of gotos can take the control to the same label.
- The label can be on a separate line or on the same line as the
- statement following it, as in,

  sos : printf ("To err is human!") ;

THANK YOU