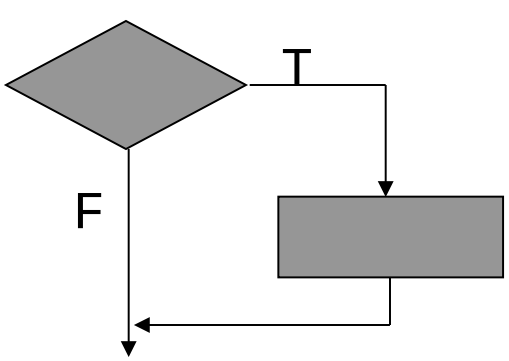
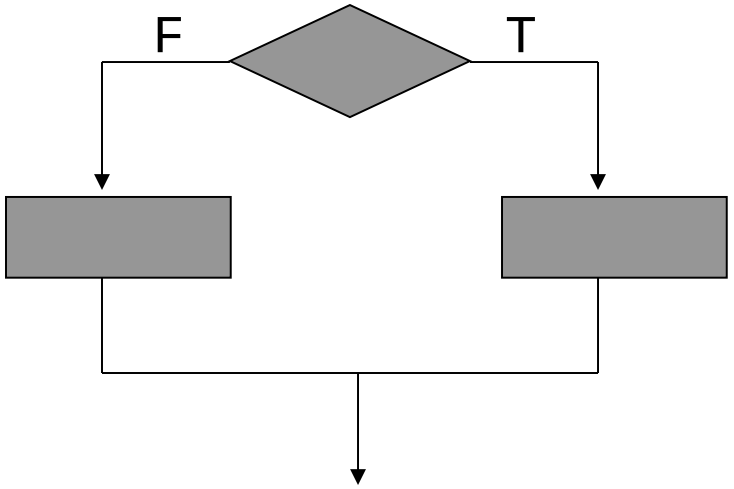
C++ Control Structures

Part II - Case

Decision Logic

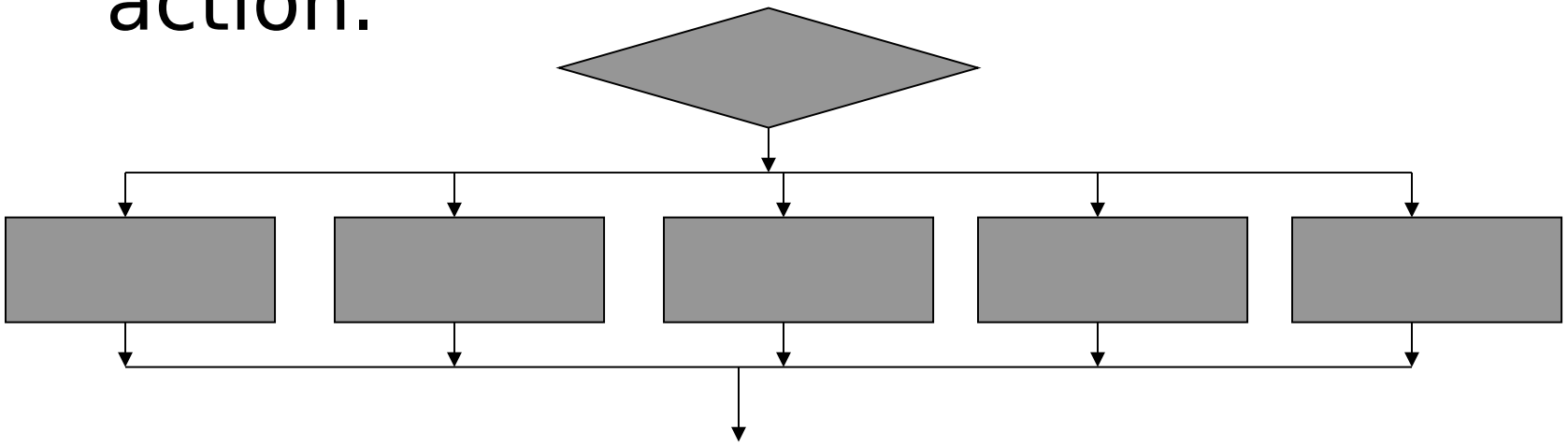
- An expression is evaluated to true or false. Depending on the value of the expression one of two paths is chosen.





Case Structure

- Based on the value of an expression, a switch statement can choose from several courses of action.



switch statements

```
switch ( integer expression ) {
   case value1:
     case actions;
   break;
   ...
   default:
     default actions;
}
```

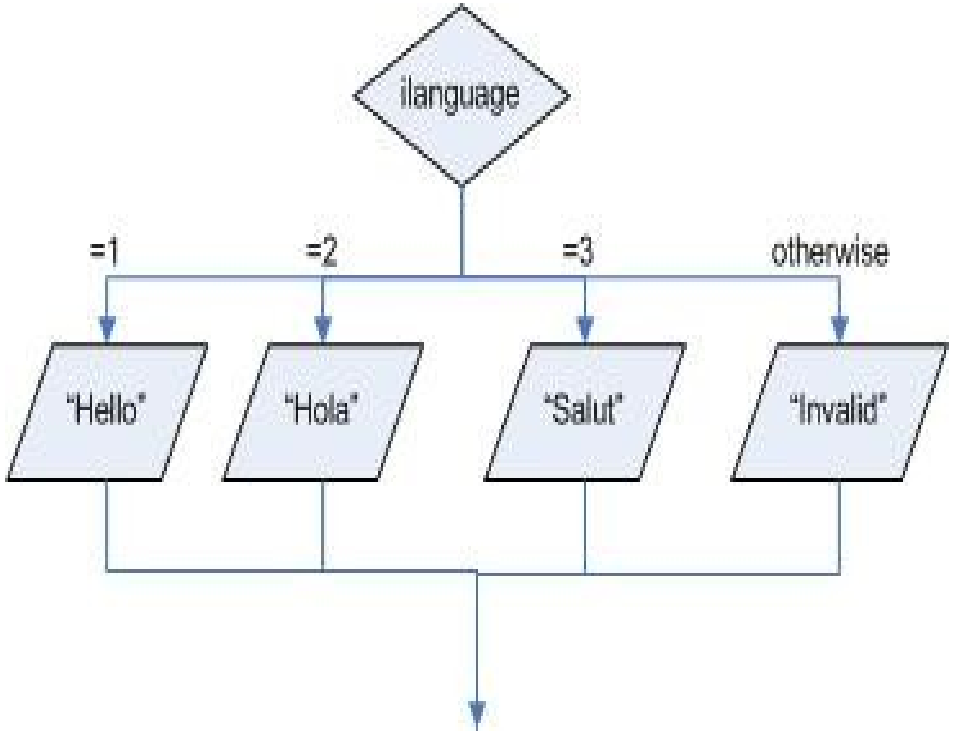
switch statements

- The integer expression can be a character (ASCII value is used)
- When a break command is encountered, the current block is exited.
- The default case is optional

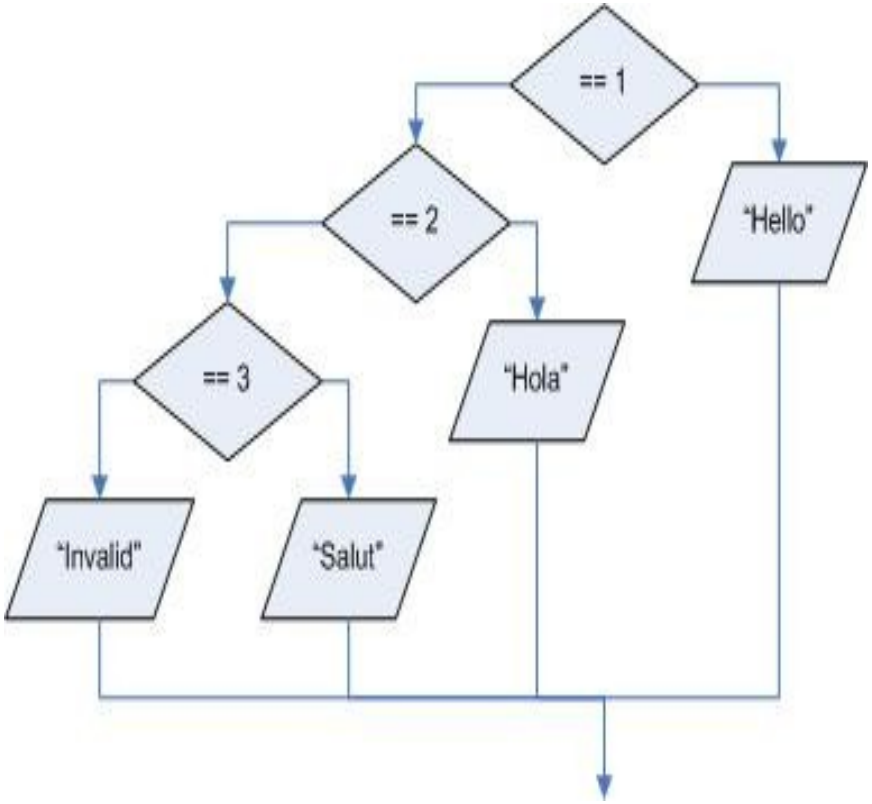
switch statements

- Each case represents a single value (switch statements are not designed to test for ranges of values)
- Good for “menu driven” scenarios
- Any valid command or structure can be placed in any case (decisions, switch statements, loops, etc)

```
switch ( ilanguage ) {
  case 1:
    cout << “Hello”;
    break;
  case 2:
    cout << “Hola”;
    break;
  case 3:
    cout << “Salut”;
    break;
  default:
    cout << “Invalid selection”;
}
```



```
if ( ilanguage == 1 )
  cout << “Hello”;
else
  if ( ilanguage == 2 )
    cout << “Hola”;
  else
    if ( ilanguage == 3 )
      cout << “Salut”;
    else
      cout << “Invalid selection”;
```



```
switch ( clanguage ) {
  case ‘e’:
  case ‘E’:
    cout << “Hello”;
    break;
  case ‘s’:
  case ‘S’:
    cout << “Hola”;
    break;
  case ‘f’:
  case ‘F’:
    cout << “Salut”;
    break;
  default:
    cout << “Invalid selection”;
}
```