ሁኔታዊ ቃላት (Conditional Statements)

4.1 ምዕላደ - ቃላት

አማርኛ ቃል	እንግሊዘኛ ቃል
ጐደሎ ቁጥሮች	real numbers
ድፍን ቁጥሮች	integer numbers
ተገማሽ ቁጥሮች	even numbers
ኢተገማሽ ቁጥሮች	odd numbers
ተፈታኝ ሁኔታ	test condition
አዎ/እውን	true
አይደለም/ሐሰት	false

4.2 የሁኔታዊ ቃላት ተግባራት ምንድን ናቸው?

ሁኔታዊ ቃላት አንድ ወይም ካንድ በላይ ተፈታኝ ሁኔታዎች በመገምገም ውሳኔ ለመስጠት የሚያስችሉ መመሪያ ቃላት ናቸው። ሁኔታዎችን በግልጽና በማያሻማ መንገድ መርምሮና ውሳኔ ላይ ደርሶ ተፈላጊ እርምጃዎችን መውሰድ ይቻላል። ሁኔታዊ ቃላት «ውሳኔ አስወሳኝ ቃላት» ተብለው በደፈናው ይጠቀሳሉ።

መመዘን የሚሹ ተጨባጭ ሁኔታዎች ሥፍር ቁጥር የላቸውም። በፕሮግራም ዓለም ዴታዎችን መዝኖ ፥ የተጠቃሚውን ፍላጐት ተገንዝቦ ፥ ያለን የጉልበትና የብቃት መጠን ለክቶ ፥ የመነጩ ውጤቶችን መርምሮ ልዩ ልዩ እርምጃ መውሰድ ዘወትራዊ ነው።

ለምሳሌ አንድ ቀላል ችግር ብንመለከት ፤ አንድ ቁጥር «ድፍን» ወይም «ጐደሎ» መሆኑን እንዴት መወሰን እንችላለን? አንዱ መንገድ ይህ ሊሆን ይችላል።

የif ቃል

```
float number = 2.5 ;

if (number == (int) number)
   System.out.println("It is whole number") ;
else
   System.out.println("It is not whole number") ;
```

ሁኔታውን በዝርዝር እንመልከት። የnumber ዕሴት 2.5 ነው። የ «(int) number» ግን 2 ነው። አንድን ተግባራዊ ቁጥር ወደ ሙሉ ቁጥር ስንቀይር የተሸራፊው ክፍል ቁጥር ይጣላል። የሁለቱ ዕሴት ለእኩልነት ስናነጻጽር ፥ ውጤቱ «አይደሉም» (ሐሰት) ይሆናል። ስለዚህ «**It is not whole number**» የሚለው ቃል ይጻፋል። የnumberን ዕሴት እንቀይርና እንደገና

እንሞክር።

የif ቃል

```
float number = 2.0 ;

if (number == (int) number)
   System.out.println("It is whole number") ;
else
   System.out.println("It is not whole number") ;
```

የnumber ሆነ የ«(int) number» ዕሴት 2 ነው። በመሆኑም ለእኩልነት ንጽጽሩ ሲካሄድ ውሴቱ «አዎ» (እውን) ይሆናል። ስለዚህ «**It is a whole number**» የሚለው ቃል ይጻፋል።

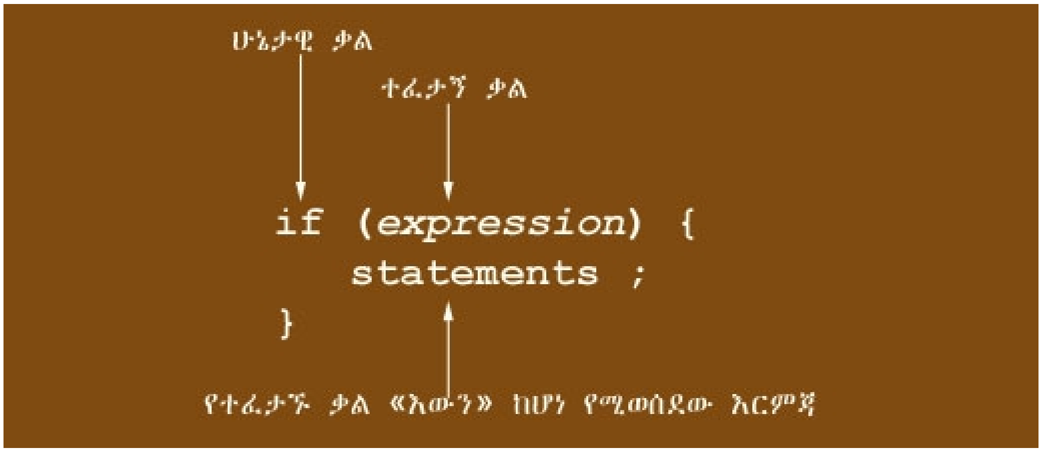
4.3 ሁኔታዊ ቃላት እነማን ናቸው?

የሁኔታዊ ቃላት አይነትና ብዛት እንደሚከተለው ነው።

4.3.1 የif-then ቃል

የif ቃል ሁል ጊዜ እንኳን ባይሆንም ሦስት አበይት ክፍሎች አሉት። 1ኛ) የif ቃል 2ኛ) ተፈታኝ ሁኔታ ወይም ሁኔታዎች 3ኛ) እርምጃ ወሳጅ ቃላት (መመሪያዎች) ናቸው።

የif-then ቃል አገባብ



ምሳሌ፦ ሁለት ኢንትጀር ተውላጠ-ቃላት **(two int variables)** አሉን እንበል። እነሱም **x** እና **y** ናቸው። ዓላማችን **x** ከ **y** ያንሳል ወይስ አያንስም የሚለውን ጥያቄ የif-thenን ቃል በመጠቀም መወሰን ነው። ቃሉ ይኸን ይመስላል። ተግባሩ ፥ የ**x** ዕሴት ከ**y** የሚያንስ ከሆነ ፥ ያንን ገልጸህ ጻፍ ነው።

የif ቃል

```
if (x < y) {
   System.out.println(x + " is less than " + y) ;
```

```
    }
```

በጥብቅ ልንረዳው የሚገባ አንዱ ነገር ቢኖር ፥ የመፈተኛ ቃላት ግምገማ ውጤት ሁልጊዜ «አዎ» ወይም «አይደለም» ነው። በሌላ አነጋገር «እውን» ወይም «ሐሰት» ነው። ከዚህ ምሳሌ አንጻር ፥ ውጡቱ «አዎ» ከሆነ በሥሩ ያለው እርምጃ ይወሰዳል። ካልሆነ ግን ፥ ያ እርምጃ አይወሰደም።

ከላይ ያየነው የif-then ቃል አጻጻፍ ለየት ባለ መልክ ሊቀርብ ይችላል። በሰውነቱ ክፍል ያለው ቃል አንድ ብቻ ከሆነ ፥ የጥምዝ-ቅንፎችን የማስገባት ግዳጅ የለም። እናም ውጤቱ ይኸን ይመስላል።

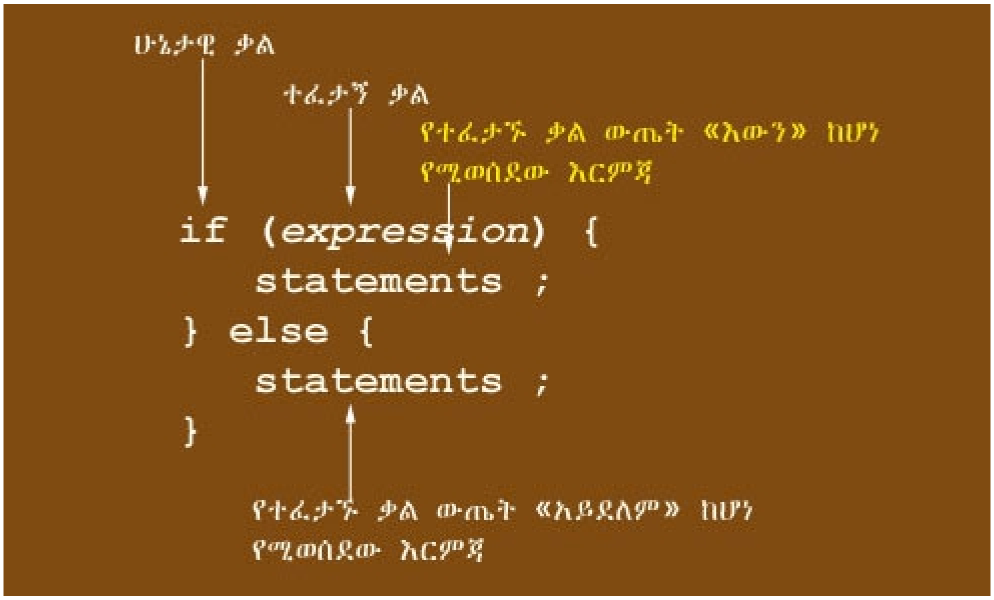
የif ቃል

```
    if (x < y)
       System.out.println(x + " is less than " + y) ;
```

4.3.2 የif-then-else

አንድ ሁኔታ ወይም ብዙ ሁኔታዎች ተመዝነው ውጤቱ «አዎ» ከሆነ ፥ በif ሥር ያለው እርምጃ ይወሰዳል። ነገር ግን ውጤቱ «አይደለም» ከሆነ በelse ሥር ያለው እርምጃ ይወሰዳል። በድጋሚ ለማሳሰብ ፥ የተፈታኝ ሁኔታ ቃል ውጤት ከ«አዎ» ወይም ከ«አይደለም» አይወጣም።

የif-then-else ቃል አገባብ



ከዚህ በላይ ያየነውን ምሳሌ ከለውጡ ጋራ እንደገና እዚህ እንመልከት።

የif ቃል

```
if (x < y) {
   System.out.println(x + " is less than " + y) ;
} else {
   System.out.println(x + " is not less than " + y) ;
}
```

ልዩነቱ ለሁለቱም ሁኔታዊ ውጤቶች ተወሳጅ እርምጃዎች አሉ። የተፈታኙ ቃል ውጤት አዎ ከሆነ በif ሥር ያለው እርምጃ ቦታውን ይወስዳል። ያ ሐሰት ከሆነ ግን በelse ሥር ብቻ ያለው እርምጃ ይፈጽማል። እንደ ሁኔታው ተገቢውን ቃል በተግባር ለማዋል ቻልን ማለት ነው።

ካንድ በላይ ተፈታኝ ሁኔታዎች ካሉን ፥ አጻጻፉና አጠቃቀሙ እንዲህ ነው። ለምሳሌ ይረዳን ዘንድ ፥ አሁንም ከላይ ያየነውን ምሳሌ እንጠቀማለን። ሦስት ሁኔታዎችን እንፈትናለን፦ ለመሆኑ x እና y እኩል ናቸው? x ከy ያንሳል? ወይም y ከx ይበልጣል?

የif ቃል

```
if (x == y)
   System.out.println(x + " is equal to " + y) ;
else if (x < y)
   System.out.println(x + " is less than " + y) ;
else if (x > y)
   System.out.println(x + " is grater than " + y) ;
```

በዚህ ቃል ሦስት ተፈታኝ ሁኔታዎች ይገመገማሉ። በሁለት ቊጥሮች መካከል ያለ ግኑኝነት ከዚህ ስለማይወጣ ፥ የአንዱ ውጤት እውን መሆኑ አይቀርም። አገማገሙ እንዲህ ነው። የመጀመሪያው «ተፈታኝ ቃል» ተነጻጽሮ ውጤቱ «አዎ» ከሆነ ፥ በሰውነቱ ያለው መመሪያ ቃል **(statement)** በሥራ ላይ ይውላል። ውጤቱ «አይደለም» ከሆነ ፥ ሁለተኛው «ተፈታኝ ቃል» ይነጻጸርና ውጠቴ «አዎ» ከሆነ ፥ በሰውነቱ ያለው መመሪያ በሥራ ይውላል። ውጤቱ «አይደለም» ከሆነ ግን ሦስተኛው «ተፈታኝ ቃል» ይመረመርና እንደ ውጤቱ ተከታዩ እርምጃ ይወሰዳል። ለተግባራዊ ምሳሌ ፕሮግራም ???? ተመልከት።

Conditional.java

```
import java.io.* ;
public class Conditional {

   InputStreamReader isr = new InputStreamReader(System.in) ;
   StreamTokenizer st = new StreamTokenizer(isr) ;

   /* Read from a keyboard and determine the data type */
   void readKBD() throws Exception {
      // prompt the user
      System.out.println("Enter your input and press ENTER") ;

      // process the keyboard input
      int state = st.nextToken() ;
      st.eolIsSignificant(true) ;

      while (state != st.TT_EOL) {
         if (state == st.TT_NUMBER)
```

```
            System.out.println("Number: " + st.nval) ;
         else if (state == st.TT_WORD)
            System.out.println("String: " + st.sval) ;
         else if (state == st.ttype)
            System.out.println("Special Character: " + (char)st.ttype) ;
         state = st.nextToken() ;
      }
   }

        /** An entry point for program execution */
   public static void main(String[] args) throws Exception {
      Conditional cond = new Conditional() ;
      cond.readKBD() ;
   }
}
```

Download: Conditional.java

4.3.3 ራሱን አቀፍ የif ቃል

ማንኛውም የif ቃል ሰውነት ሌሎች የif ቃላትን መሸከም ይችላሉ። በውስጡ ያሉት የif ቃላት እንዲሁ ሌሎችን ማቀፍ ይችላሉ። እንደዚህ አይነቱ ሁኔታዊ ቃል በእንግሊዘኛ «**nested if**» ቃል ይሉታል።

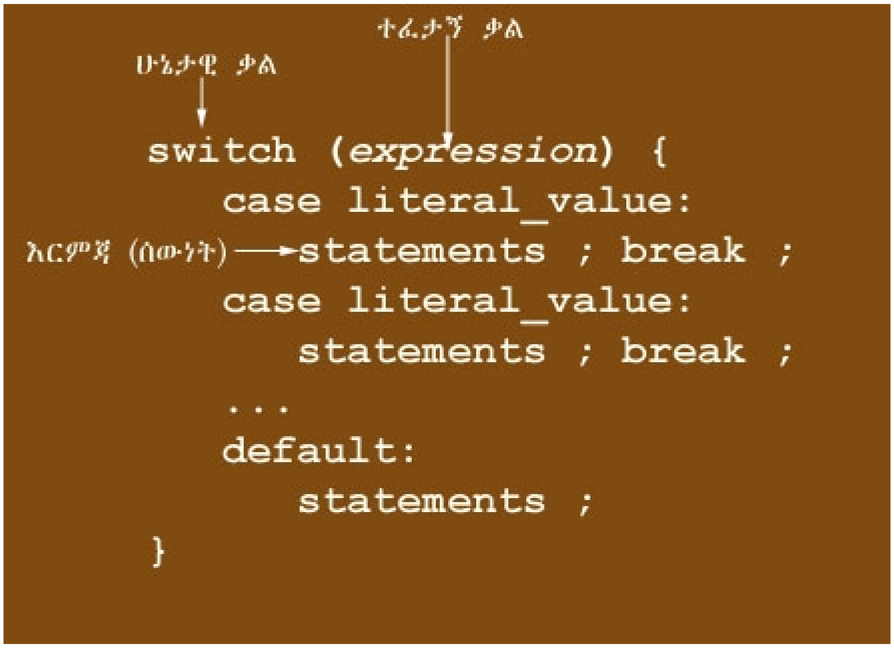
የif ቃል

```
     ...
     String state = request.getParameter("state") ;
     if ( (state != null) && (!state.equals("")) ) {
        if (state.equals("TX")
           // Tax applies
        else
           // Tax doesn't apply
     } else {
        // do something here
     }
```

4.4 የswitch ቃል

የswitch ቃል ሁኔታዊ ቃል ነው። በቃል-አገባብ ፥ በአሠራር ፥ እንዲሁም በጠቀሜታ ከif ይለያል። ቃሉ አንድን «ተፊታኝ-ቃል» **(expression)** በተረታ ከተሰጡ ልዩ ልዩ አማራጮች ጋር በማነጻጸርና ከየትኛው ጋር እኩል እንደሆነ የሚያፈላልግና እርምጃ የሚያስወስድ ነው።

የswitch ቃል አገባብ ይኸን ይመስላል።



ከዚህ በታች የቀረበው ምሳሌ ፥ የ«dayን ዕሴት ከፈለቀቀ በኋላ ከያንዳንዱ የcase ዕሴት ጋራ እኩልነት እስከተገኘ ድረስ ያነጻጽራል። የግምገማው ውጤት «አዎ» ካልሆነ የdefault ክፍል ሥራ ላይ ይውላል።

ተፈታኙ ቃል «day» ሲሆን የሚነጻጸረው ከያንዳንዱ የcase ቃል ጋር ሆኖ ፥ እኩል የሆነው በሥራ ይውልና ሌሎቹ ይዘለላሉ። በswitch ሕግ መሠረት የሚነጻጸሩት አማራጮች የግድ በራሳቸው የቆሙ ዕሴት መሆን አለባቸው። ታውላጠ-ቃል አይፈቀደም።

የswitch ቃል

```
byte day = random.nextInt(6) + 1 ;

switch(day) {
   case 1:
      System.out.println("Sunday");    break ;
   case 2:
      System.out.println("Monday") ;   break ;
   case 3:
      System.out.println("Tuesday") ;  break ;
   case 4:
      System.out.println("Wednesday"); break ;
   case 5:
      System.out.println("Thursday") ; break ;
   case 6:
      System.out.println("Friday") ;   break ;
   default:
      System.out.println("Saturday") ; break ;
}
```

የእያንዳንዱ አማራጭ ሰውነት ለብቻው በሥራ ላይ እንዲውል ከተፈለገ በbreak ቃል መገደብ አለበት። ለዚህም ነው በያንዳንዱ ክፍል የbreak እላይ የታከለው። ያንን ያለማድረግ ግን ቀጥሎ በተሰጠው ምሳሌ ያለውን ሁኔታ ያመጣል።

የif ቃል

```
switch(number) {
   case 16:
      System.out.print(number * 1) ;
      System.out.print(", ") ;
   case 32:
      System.out.print(number * 2) ;
      System.out.print(", ") ;
   case 64:
      System.out.print(number * 4) ;
      System.out.print(", ") ;
   case 128:
      System.out.print(number * 8) ;
      System.out.print(", ") ;
   default:
      System.out.print(number * 16) ;
}
```

ይህ ቃል በሥራ ላይ ከዋለ ፥ የሚከተለው ይሆናል። ለምሳሌ ያህል የnumber ዕሴት **64** ነበር። ከመጀመሪያው የcase ቃል ማለትም **16** ይነጻጸራል። ውጤቱ እኩል አይደሉም ነው። ቀጥሎ ከ32 ጋራ ይነጻጸራል ፤ አሁንም ውጤቱ እኩል አይደሉም ነው። ነገር ግን ከ64 ጋረ ሲነጻጸሩ እኩል ስለሚሆኑ ፥ በሥሩ ያለው እርምጃ ይወሰዳል። ነገር ግን የእርምጃው ቃል በbreake ቃል ስላልተገደበ ፥ ቀጥለው ያሉት የሰውነት ክፍሎች በሙሉ በሥራ ላይ ይውላሉ። በዚህ የተነሳ የሚወጣው ውጤት ይኸን ይመስላል።

64, 128, 256

ይህ የswitch መሠረታዊ ባህሪ ነው። ላንዳንድ ችግሮች ይህ ጥሩ መፍትሔ ይሆናል። ይኸን ባህሪ ለማረም መድኃኒቱ በእያንዳንዱ የሰውነት መጨረሻ ላይ የbreakን ቃል ማስገባት ነው።

Switch.java

```
import java.util.* ;
public class Switch {

   /* Given an ineger, it returns the day of the week */
   String day(int n) {
      switch(n) {
         case 1:
                return "Sunday" ;
         case 2:
                return "Monday" ;
         case 3:
                return "Tuesday" ;
         case 4:
                return "Wednesday" ;
         case 5:
                return "Thursday" ;
         case 6:
                return "Friday" ;
         default:
                return "Saturday" ;
       }
```

```
    }

    /** An entry point for program execution */
    public static void main(String[] args) throws Exception {
         Random rand = new Random() ;
         Switch cond = new Switch() ;

         int i = rand.nextInt(7) + 1 ;
         System.out.println( cond.day(i)) ;
    }
}
```

Download: Switch.java

To contact: info@senamirmir.com

Copyright © 2002-2005 Senamirmir Project