

# **CS200 – Програмчлалын үндэс**

Лекц 03

С хэлний програмын удирдлага

Профессор А.Эрдэнэбаатар

# Лекцийн агуулга

---

- ▶ **while, for, do...while** давталтын операторууд
- ▶ Програмын удирдлагын урсгалыг өөрчлөгч **break, continue** операторууд
- ▶ **switch** сонголтын оператор
- ▶ Логик үйлдэл ба удирдлагын операторын нийлмэл нөхцөл
- ▶ Тэнцэтгэл болон олгох үйлдлүүд
- ▶ Бүтцээр програмчлах асуудалд
- ▶ Дүгнэлт

# Давталтын тухай

---

- ▶ Давталт
  - ▶ Компьютер ямар нэг нөхцөл **true** байвал бүлэг удирдамжийг давтан гүйцэтгэсээр байна
- ▶ Тоолууран удирдлагатай давталт
  - ▶ Тодорхой давталт: хэдэн удаа давтах нь илэрхий байх
  - ▶ Давталтыг тоолохдоо удирдах хувьсагч ашигладаг
- ▶ Онцгой утган удирдлагатай давталт
  - ▶ Тодорхойгүй тооны давталт
  - ▶ Давталтын тоо тодорхойгүй үед ашигладаг
  - ▶ Онцгой утга нь “өгөгдлийн төгсгөл” –г илэрхийлдэг

# Тоолууран удирдлагатай давталт

---

- ▶ Тоолууран удирдлагатай давталт дараахыг шаардана
  - ▶ Удирдах хувьсагч –н нэр (эсхүл давталтын тоолуур)
  - ▶ Удирдагч хувьсагчийн анхны утга
  - ▶ Давталт бүрт удирдах хувьсагчийг нэмэгдүүлэх (хорогдуулах) –аар өөрчлөх
  - ▶ Удирдах хувьсагчийн сүүлчийн утга эсэхийг шалгах нөхцөл (ө.х. давталт үргэлжиллэх эсэхийг шалгах)

# Тоолууран удирдлагатай давталт

---

## ► Жишээ:

```
int counter = 1;           /* initialization */
while ( counter <= 10 ) { /* repetition condition */
    printf( "%d\n", counter );
    ++counter;             /* increment */
}
```

## ► Эхний оператор

```
int counter = 1;
```

- **counter** удирдах хувьсагчийн нэр
- Түүнийг бүхэл гэж тодорхойлжээ
- Түүнд санах ойд орон зай гаргана
- Анхны утга нь **1**

# Тоолууран удирдлагатай давталт

```
1.  /* Ex_10 Counter-controlled repetition */
2.  #include <stdio.h>
3.
4.  /* function main begins program execution */
5.  int main( void )
6.  {
7.      int counter = 1; /* initialization */
8.
9.      while ( counter <= 9 ) { /* repetition condition */
10.         printf ( "%d\n", counter ); /* display counter */
11.         counter++; /* increment */
12.     } /* end while */
13.
14.     return 0; /* indicate program ended successfully */
15.
16. } /* end function main */
```

Тодорхойлолт болон утга  
олголтыг зэрэг хийж байна

1  
2  
3  
4  
5  
6  
7  
8  
9  
10

# Тоолууран удирдлагатай давталт

---

## ► Хураангуйлсан код

- Програм зохиогчид кодоо илүү хураангуй болгохыг боддог

```
int counter = 0;  
while ( ++counter <= 10 )  
    printf( "%d\n", counter );
```

## ► Чухал санамж!

Давталтын тоолуурт хөвөгч таслалтай хувьсагчийг ашигласнаар тоолуурын утга яг нарийн биш болж болзошгүй бөгөөд төгсгөх шалгалт буруу болж болох юм. Учир нь хөвөгч таслалтай утга компьютерт ойролцоогоор дүрслэгддэг

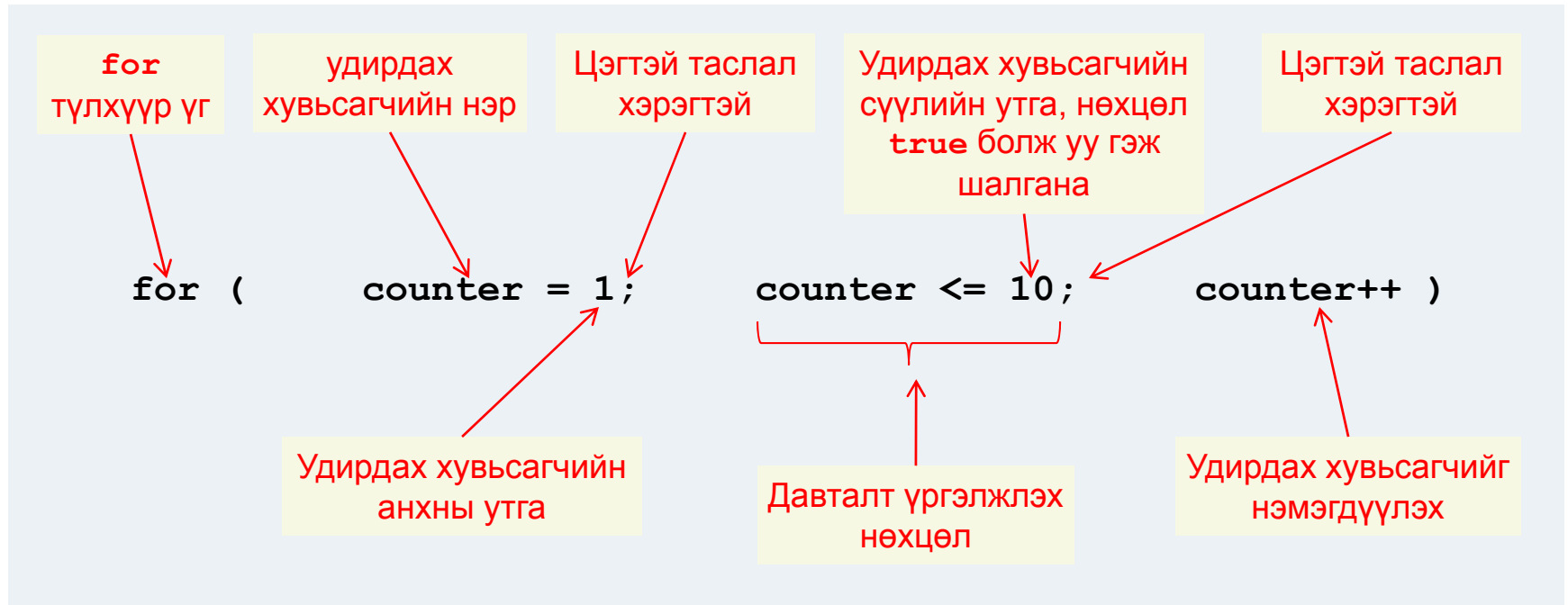
# Тоолууран удирдлагатай for давталт

```
1.  /* Ex_11 Counter-controlled repetition with the for statement */
2.  #include <stdio.h>
3.
4.  /* function main begins program execution */
5.  int main( void )
6.  {
7.      int counter; /* define counter */
8.
9.      /* initialization, repetition condition, and increment
10.         are all included in the for statement header. */
11.      for ( counter = 1; counter <= 10; counter++ ) {
12.          printf( "%d\n", counter );
13.      } /* end for */
14.
15.      return 0; /* indicate program ended successfully */
16.
17. } /* end function main */
```

for давталт counter –г 1 гэж эхлээд <= 10 байгаа бол давтсаар байна. Давталтын төгсгөлд ирэх бүр counter –н утга 1 –ээр нэмэгдэнэ



# for давталтын оператор



# for давталтын оператор


## ► for давталтыг бичих загвар

```
for ( initialization; loopContinuationTest; increment )  
    statement
```

## ► Жишээ:

```
for ( counter = 1; counter <= 10; counter++ )  
    printf( "%d\n", counter );
```

## ► 1 –ээс 10 хүртэлх тоог хэвлэж байна



Сүүлийн илэрхийллийн  
ард цэгтэй таслал байхгүй  
байгааг анхаар

# for оператор

---

- ▶ **for** давталтыг **while** оператор ашиглаад өөрөөр бичиж болдог

*initialization;*

```
while ( loopContinuationTest ) {  
    statement;  
    increment;  
}
```

- ▶ Идэвхижүүлэлт ба нэмэгдүүлэлт

- ▶ Таслалаар тусгаарлагдсан жагсаалт байж болно

- ▶ Жишээ:

```
for ( i = 0, j = 0; j + i <= 10; j++, i++ )  
    printf( "%d\n", j + i );
```

# for оператор: санамж, зөвлөмж

## ► Арифметик илэрхийлэл

- Идэвхижүүлэлт, давталтын нөхцөл, нэмэгдүүлэлт арифметик илэрхийллийг агуулж болно. Хэрвээ  $x = 2$ ,  $y = 10$  бол

```
for ( j = x; j <= 4 * x * y; j += y / x )
```

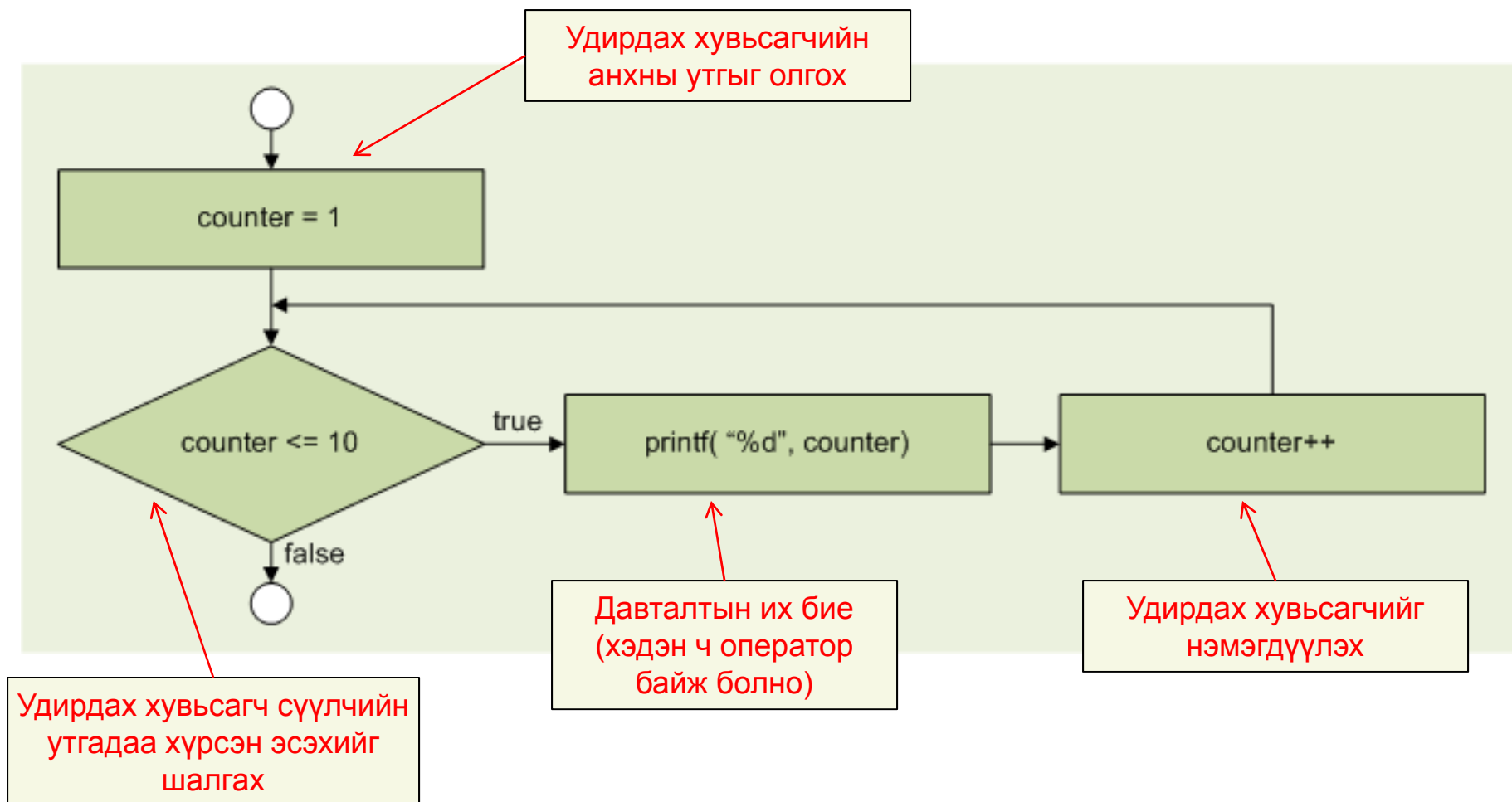
гэдэг нь дараахтай ижил

```
for ( j = 2; j <= 80; j += 5 )
```

## ► **for** операторын талаарх санамж

- “Нэмэгдүүлэлт” сөрөг байж болно (хорогдуулалт)
- Хэрвээ давталт үргэлжлэх нөхцөл анхнаасаа **false** байсан бол их биений оператор биелэгдэхгүй, удирдлага **for** операторын дараачийн операторт шилжинэ
- Операторын их биед удирдах хувьсагчийг голдуу хэвлэдэг, гэхдээ заавал биш
- Хэдийгээр удирдах хувьсагчийн утгыг операторын их биед өөрчилж болох ч алдаа дагуулах магадлалтай. Иймд өөрчлөхгүй байвал дээр

# for давталтын операторын нийтлэг блок схем



```

1.  /* Ex_12 Calculating compound interest */
2.  #include <stdio.h>
3.  #include <math.h>
4.
5.  /* function main begins program execution */
6.  int main( void )
7.  {
8.      double amount;          /* amount on deposit */
9.      double principal = 1000.0; /* starting principal */
10.     double rate = .05;       /* annual interest rate */
11.     int year;                /* year counter */
12.
13.     /* output table column head */
14.     printf( "%4s%21s\n", "Year", "Amount on deposit" );
15.
16.     /* calculate amount on deposit for each of ten years */
17.     for ( year = 1; year <= 10; year++ ) {
18.
19.         /* calculate new amount for specified year */
20.         amount = principal * pow( 1.0 + rate, year );
21.
22.         /* output one table row */
23.         printf( "%4d%21.2f\n", year, amount );
24.     } /* end for */
25.
26.     return 0; /* indicate program ended successfully */
27.
28. } /* end function main */

```

Өөр нэг толгой файл.

pow функц эхний аргументыг хоёр дахь аргументээр зэрэгт дэвшүүлдэг. Функц double төрлийн аргументуудтай байх ба мөн ийм төрлийн утга буцаана.

# for операторын өмнөх жишээний хариу

Year	Amount on deposit
1	1050.00
2	1102.50
3	1157.63
4	1215.51
5	1276.28
6	1340.10
7	1407.10
8	1477.46
9	1551.33
10	1628.89

# do...while давталтын оператор

---

- ▶ **do...while** давталтын оператор
  - ▶ **while** бүтэцтэй төстэй
  - ▶ Давталтын нөхцлийг давталтын их бие биелэгдсэний дараа шалгадаг. Иймд их бие ядаж 1 удаа биелэгдэнэ
- ▶ Операторыг бичих загвар

```
do {  
    statement;  
} while ( condition );
```

- ▶ Жишээ: **counter = 1** гэж бодьё

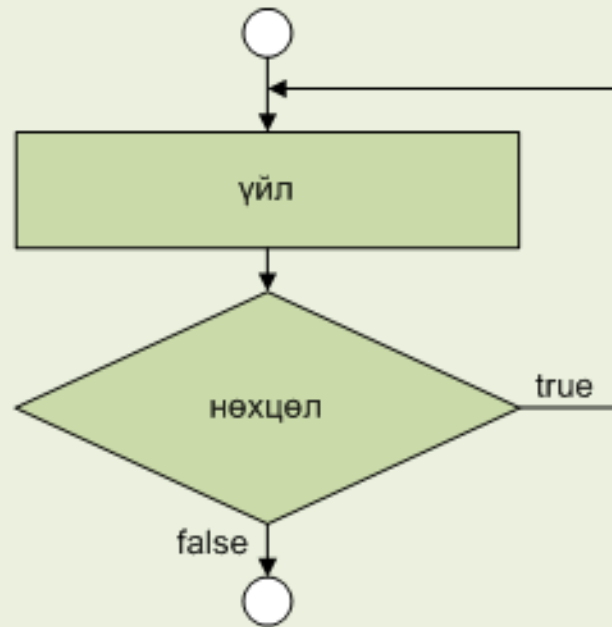
```
do {  
    printf( "%d ", counter );  
} while ( ++counter <= 10 );
```

- ▶ 1 –ээс 10 хүртэлх бүхэл тоог хэвлэж байна



# do...while давталтын операторын блок схем

---



# do...while давталтын жишээ

```
1.  /* Ex_13 Using the do/while repetition statement */
2.  #include <stdio.h>
3.
4.  /* function main begins program execution */
5.  int main( void )
6.  {
7.      int counter = 1;                /* initialize counter */
8.
9.      do {
10.         printf( "%d  ", counter ); /* display counter */
11.     } while ( ++counter <= 10 ); /* end do...while */
12.
13.     return 0; /* indicate program ended successfully */
14.
15. } /* end function main */
1 2 3 4 5 6 7 8 9 10
```

counter –г нэмэгдүүлээд  
дараа нь 10 –аас бага буюу  
тэнцүү эсэхийг шалгаж байна

# Лекцийн агуулга

---

- ▶ **while, for, do...while** давталтын операторууд
- ▶ Програмын удирдлагын урсгалыг өөрчлөгч **break, continue** операторууд
- ▶ **switch** сонголтын оператор
- ▶ Логик үйлдэл ба удирдлагын операторын нийлмэл нөхцөл
- ▶ Тэнцэтгэл болон олгох үйлдлүүд
- ▶ Бүтцээр програмчлах асуудалд
- ▶ Дүгнэлт

# break, continue операторууд

---

## ▶ break

- ▶ while, for, do...while, switch операторуудаас шууд гарна
- ▶ Дээрх бүтцүүдийн дараах эхний операторт програмын удирдлага шилжинэ
- ▶ break операторыг хэрэглэх ерөнхий тохиолдол
  - ▶ Давталтаас эрт гарах
  - ▶ switch операторын үлдсэн хэсгийг алгасах

# break операторын жишээ

```
1.  /* Ex_14 Using the break statement in a for statement */
2.  #include <stdio.h>
3.
4.  /* function main begins program execution */
5.  int main( void )
6.  {
7.      int x; /* counter */
8.
9.      /* loop 10 times */
10.     for ( x = 1; x <= 10; x++ ) {
11.
12.         /* if x is 5, terminate loop */
13.         if ( x == 5 ) {
14.             break; /* break loop only if x is 5 */
15.         } /* end if */
16.
17.         printf( "%d ", x ); /* display value of x */
18.     } /* end for */
19.
20.     printf( "\nBroke out of loop at x == %d\n", x );
21.
22.     return 0; /* indicate program ended successfully */
23.
24. } /* end function main */
1 2 3 4
Broke out of loop at x == 5
```

break энд for давталтыг  
төгсгөж байна.

# break, continue операторууд

---

## ▶ **continue**

- ▶ **while, for, do...while** операторуудын **их биеийн үлдсэн операторуудыг алгасна**

- ▶ Өөрөөр хэлвэл давталтын дараачийн тойрогт шилжинэ

## ▶ **while, do...while**

- ▶ **continue** операторын дараа шууд давталтын төгсгөлийн шалгалт хийгдэнэ

## ▶ **for**

- ▶ Нэмэгдүүлэх илэрхийлэл биелэгдэж, дараа нь давталтын төгсгөлийн шалгалт хийгдэнэ

# continue операторын жишээ

```
1.  /* Ex_15 Using the continue statement in a for statement */
2.  #include <stdio.h>
3.
4.  /* function main begins program execution */
5.  int main( void )
6.  {
7.      int x; /* counter */
8.
9.      /* loop 10 times */
10.     for ( x = 1; x <= 10; x++ ) {
11.
12.         /* if x is 5, continue with next iteration of loop */
13.         if ( x == 5 ) {
14.             continue; /* skip remaining code in loop body */
15.         } /* end if */
16.
17.         printf( "%d ", x ); /* display value of x */
18.     } /* end for */
19.
20.     printf( "\nUsed continue to skip printing the value 5\n" );
21.
22.     return 0; /* indicate program ended successfully */
23.
24. } /* end function main */
```

continue ЭНД for  
давталтын төгсгөл  
хүртэл алгасч,  
дараачийн тойрогт  
шилжүүлж байна.

1 2 3 4 6 7 8 9 10

Use continue to skip printing the value 5

# Лекцийн агуулга

---

- ▶ **while, for, do...while** давталтын операторууд
- ▶ Програмын удирдлагын урсгалыг өөрчлөгч **break, continue** операторууд
- ▶ **switch** сонголтын оператор
- ▶ Логик үйлдэл ба удирдлагын операторын нийлмэл нөхцөл
- ▶ Тэнцэтгэл болон олгох үйлдлүүд
- ▶ Бүтцээр програмчлах асуудалд
- ▶ Дүгнэлт



# Олон-сонголтын `switch` оператор

---

## ► `switch`

- Хувьсагч болон илэрхийллийн боломжид утга бүрийг шалгаж, өөр өөр үйлийг гүйцэтгэхэд тустай

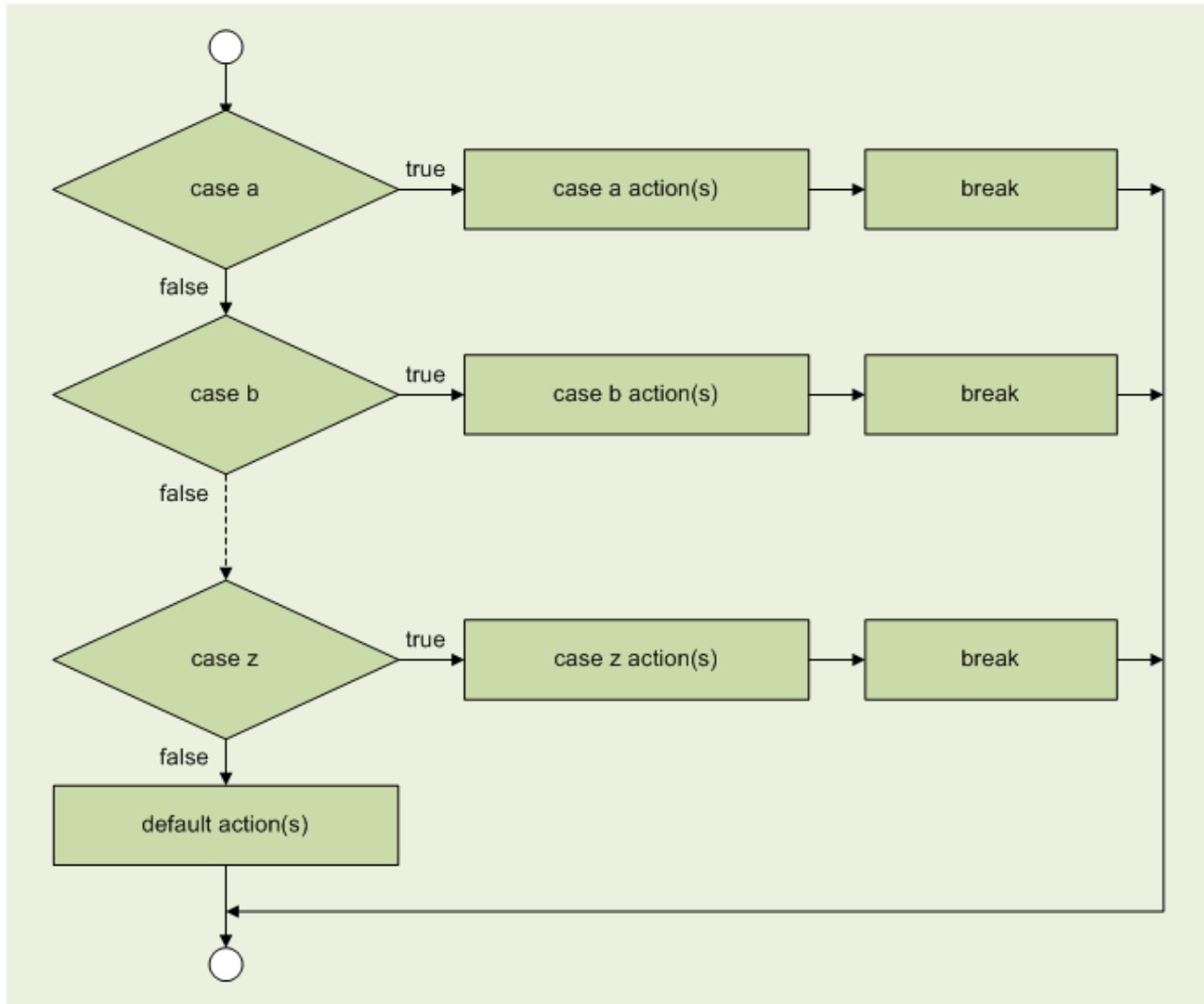
## ► Загвар

- Цуврал `case` шошго ба нэмэлт `default` тохиолдол

```
switch (value) {  
    case '1':  
        actions  
    case '2':  
        actions  
    default:  
        actions  
}
```

- Энэ операторын дотроос гарахад `break` –г ашигладаг

# switch операторын блок схем



# switch операторын жишээ

```
1.  /* Ex_16 Counting letter grades */
2.  #include <stdio.h>
3.
4.  /* function main begins program execution */
5.  int main( void )
6.  {
7.      int grade;      /* one grade */
8.      int aCount = 0; /* number of As */
9.      int bCount = 0; /* number of Bs */
10.     int cCount = 0; /* number of Cs */
11.     int dCount = 0; /* number of Ds */
12.     int fCount = 0; /* number of Fs */
13.
14.     printf( "Enter the letter grades.\n" );
15.     printf( "Enter the EOF character to end input.\n" );
16.
17.     /* loop until user types end-of-file key sequence */
18.     while ( ( grade = getchar() ) != EOF ) {
19.
20.         /* determine which grade was input */
21.         switch ( grade ) { /* switch nested in while */
22.
23.             case 'A': /* grade was uppercase A */
24.             case 'a': /* or lowercase a */
25.                 ++aCount; /* increment aCount */
26.                 break; /* necessary to exit switch */
```

EOF гэдэг нь “end of file” гэсэн үг. Энэ тэмдэгт систем бүрт өөр байдаг

switch өөрт нь үүрэлсэн бүх case –г тулгаж үздэг

break оператор switch –н төгсгөл хүртэл алгасах боломжийг програмд олгодог

# switch операторын жишээ

```
27.  
28.     case 'B': /* grade was uppercase B */  
29.     case 'b': /* or lowercase b */  
30.         ++bCount; /* increment bCount */  
31.         break; /* exit switch */  
32.  
33.     case 'C': /* grade was uppercase C */  
34.     case 'c': /* or lowercase c */  
35.         ++cCount; /* increment cCount */  
36.         break; /* exit switch */  
37.  
38.     case 'D': /* grade was uppercase D */  
39.     case 'd': /* or lowercase d */  
40.         ++dCount; /* increment dCount */  
41.         break; /* exit switch */  
42.  
43.     case 'F': /* grade was uppercase F */  
44.     case 'f': /* or lowercase f */  
45.         ++fCount; /* increment fCount */  
46.         break; /* exit switch */  
47.  
48.     case '\n': /* ignore newlines, */  
49.     case '\t': /* tabs, */  
50.     case ' ': /* and spaces in input */  
51.         break; /* exit switch */
```

# switch операторын жишээ

```
27.  
28.     default: /* catch all other characters */  
29.         printf( "Incorrect letter grade entered." );  
30.         printf( " Enter a new grade.\n" );  
31.         break; /* optional; will exit switch anyway */  
32.     } /* end switch */  
33.  
34. } /* end while */  
35.  
36. /* output summary of results */  
37. printf( "\nTotals for each letter grade are:\n" );  
38. printf( "A: %d\n", aCount ); /* display number of A grades */  
39. printf( "B: %d\n", bCount ); /* display number of B grades */  
40. printf( "C: %d\n", cCount ); /* display number of C grades */  
41. printf( "D: %d\n", dCount ); /* display number of D grades */  
42. printf( "F: %d\n", fCount ); /* display number of F grades */  
43.  
44. return 0; /* indicate program ended successfully */  
45.  
46. } /* end function main */
```

default тохиолдол нь  
бүх case тохироогүй үед  
биелэгддэг

# switch операторын жишээ

Enter the letter grades.

Enter the EOF character to end input.

a

b

c

C

A

d

f

C

E

Incorrect letter grade entered. Enter a new grade.

D

A

B

^D

Totals for each letter grade are:

A: 3

B: 2

C: 3

D: 2

F: 1

# Лекцийн агуулга

---

- ▶ **while, for, do...while** давталтын операторууд
- ▶ Програмын удирдлагын урсгалыг өөрчлөгч **break, continue** операторууд
- ▶ **switch** сонголтын оператор
- ▶ Логик үйлдэл ба удирдлагын операторын нийлмэл нөхцөл
- ▶ Тэнцэтгэл болон олгох үйлдлүүд
- ▶ Бүтцээр програмчлах асуудалд
- ▶ Дүгнэлт

# Логик үйлдлүүд

- ▶ **&&** (логик AND – “ба”)
  - ▶ Хоёр нөхцөл хоёул **true** бол гарах утга **true**
- ▶ **||** (логик OR – “буюу”)
  - ▶ Хоёр нөхцлийн ядаж нэг нь **true** бол гарах утга **true**
- ▶ **!** (логик NOT – “үгүй”)
  - ▶ Өгөгдсөн нөцлийн үнэн/худлын үгүйсгэл
  - ▶ Нэг гишүүнт үйлдэл
- ▶ **Жишээ:**

Илэрхийлэл	Үр дүн
<code>true &amp;&amp; false</code>	<code>false</code>
<code>true    false</code>	<code>true</code>
<code>!false</code>	<code>true</code>



# Логик үйлдлүүд

expression1	expression2	expression1 && expression2
0	0	0
0	nonzero	0
nonzero	0	0
nonzero	nonzero	1

&& (логик AND) үйлдлийн  
үнэний хүснэгт

expression1	expression2	expression1    expression2
0	0	0
0	nonzero	0
nonzero	0	0
nonzero	nonzero	1

|| (логик OR) үйлдлийн  
үнэний хүснэгт

expression	!expression
0	1
nonzero	0

! (логик NOT) үйлдлийн  
үнэний хүснэгт

# Үйлдлүүдийн ахлах чанар ба биелэх чиглэл

Үйлдэл	Биелэгдэх чиглэл	Төрөл
++(постфикс) --(постфикс)	Баруунаас зүүн	Постфикс
+ - ! (төрөл) ++(префикс) --(префикс)	Баруунаас зүүн	Ганц гишүүнт
* / %	Зүүнээс баруун	Үржигдэх
+ -	Зүүнээс баруун	Нэмэгдэх
< <= > >=	Зүүнээс баруун	Харьцаа
== !=	Зүүнээс баруун	Тэнцэтгэл
&&	Зүүнээс баруун	Логик AND
	Зүүнээс баруун	Логик OR
?:	Баруунаас зүүн	Нөхцөлт
= += -= *= /= %=	Баруунаас зүүн	Олгох
,	Зүүнээс баруун	Таслал

# Лекцийн агуулга

---

- ▶ **while, for, do...while** давталтын операторууд
- ▶ Програмын удирдлагын урсгалыг өөрчлөгч **break, continue** операторууд
- ▶ **switch** сонголтын оператор
- ▶ Логик үйлдэл ба удирдлагын операторын нийлмэл нөхцөл
- ▶ Тэнцэтгэл болон олгох үйлдлүүд
- ▶ Бүтцээр програмчлах асуудалд
- ▶ Дүгнэлт

# Тэнцэтгэл (==) болон олгох (=) үйлдлүүдийн будлиан

---

- ▶ Сольж хэрэглэснээс гарах ноцтой алдаа
  - ▶ Дүрмийн алдаа үүсдэггүй
  - ▶ Ямар ч илэрхийллээс утга гарч, түүнийг нь удирдлагын бүтцэд ашиглаж болдог
  - ▶ Nonzero (0 биш) утга – **true**, 0 утга – **false** байдаг
  - ▶ == ашигласан жишээ:

```
if (payCode == 4)
```

```
    printf( "You get a bonus!\n" );
```

- ▶ **payCode** –г шалгаад 4 –тэй тэнцүү бол текст хэвлэгдэнэ

- ▶ Оронд нь = ашигласан жишээ:

```
if (payCode = 4)
```

```
    printf( "You get a bonus!\n" );
```

- ▶ **payCode = 4** буюу **true** болж **payCode** ямар байхаас үл хамаарч текст хэвлэгдэнэ. Ө.х. **дүрмийн биш, логик алдаа**

# Тэнцэтгэл (==) болон олгох (=) үйлдлүүдийн будлиан

---

## ► lvalues (зүүн утга)

- Тэгшитгэлийн зүүн талд байж болох илэрхийлэл
- Түүний утгыг хувьсагчийн нэр шиг өөрчилж болно

**x = 4;**

## ► rvalues (баруун утга)

- Тэгшитгэлийн зөвхөн баруун талд байж болох илэрхийлэл
- Тухайлбал, тоо мэтийн тогтмолууд

**4 = x;** гэж болохгүй харин **x = 4;** гэж бичнэ

- lvalues –г rvalues болгон ашиглаж болно. Харин эсрэгээр болохгүй

**y = x;**

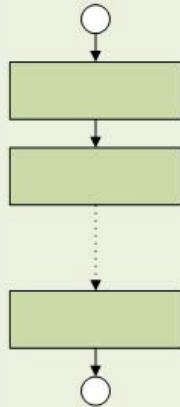
# Лекцийн агуулга

---

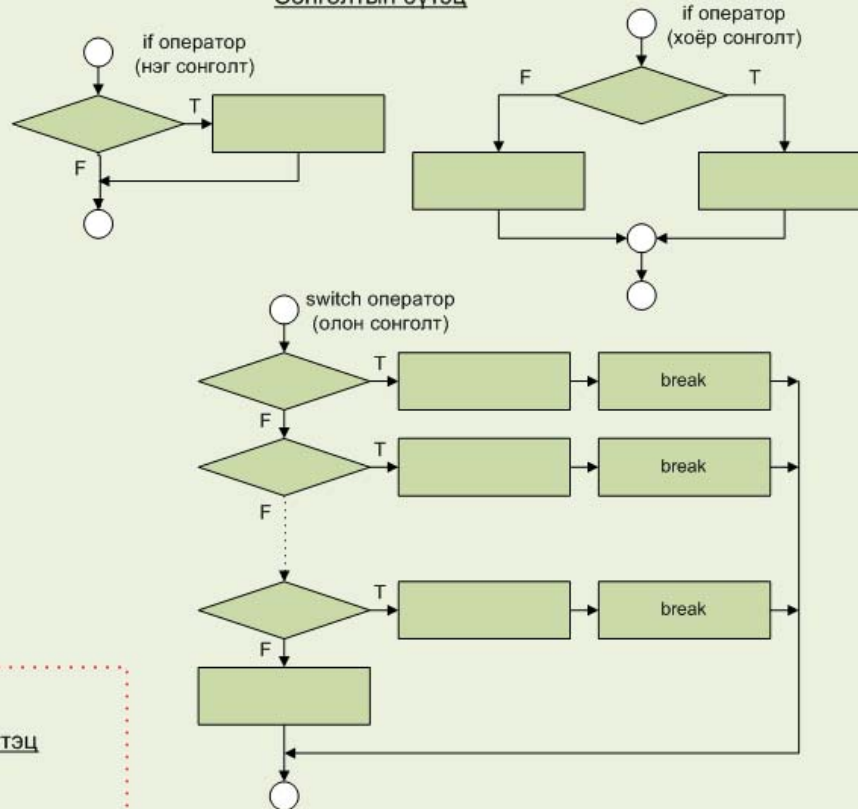
- ▶ **while, for, do...while** давталтын операторууд
- ▶ Програмын удирдлагын урсгалыг өөрчлөгч **break, continue** операторууд
- ▶ **switch** сонголтын оператор
- ▶ Логик үйлдэл ба удирдлагын операторын нийлмэл нөхцөл
- ▶ Тэнцэтгэл болон олгох үйлдлүүд
- ▶ Бүтцээр програмчлах асуудалд
- ▶ Дүгнэлт

## Бүтцүүд:

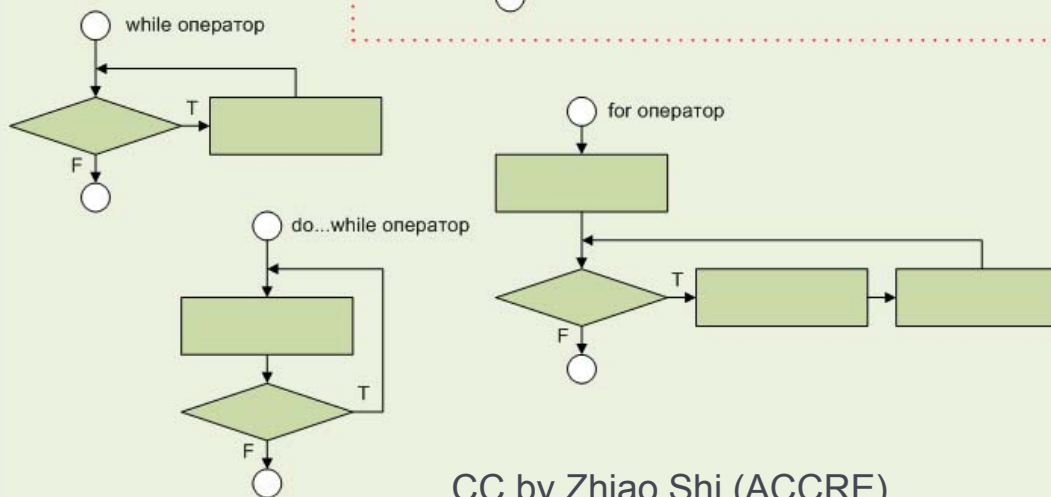
### Дарааллын бүтэц



### Сонголтын бүтэц



### Давталтын бүтэц



# Бүтцээр програмчлах асуудалд

---

## ▶ Бүтцийн програмчилал

- ▶ Бүтэцлэгдээгүй програмтай харьцуулахад програмыг засах, зүгшрүүлэх, шалгах ажиллагаа хялбар

## ▶ Бүтцээр програмчилах дүрэм

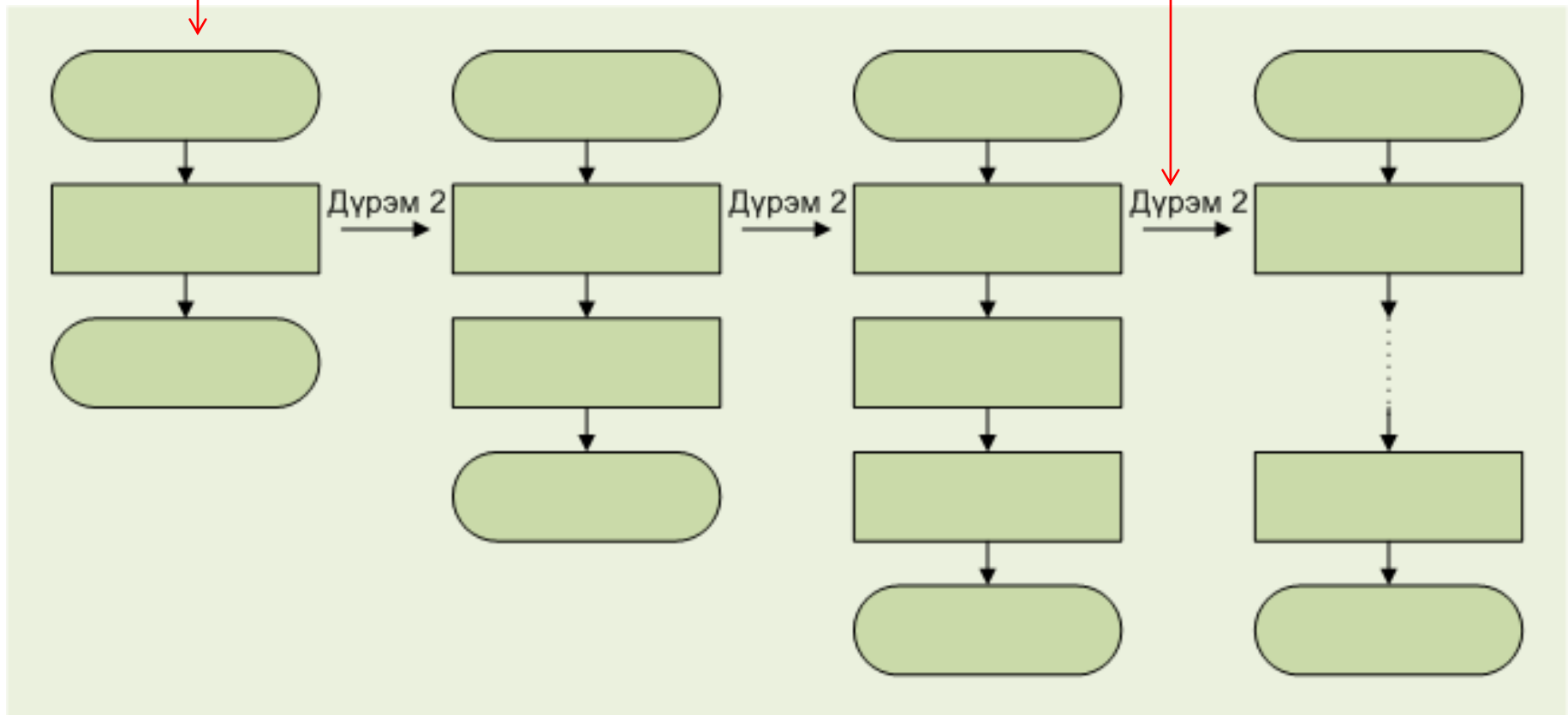
- ▶ Дүрмүүд програмчилалын хөгжлөөр бий болсон
- ▶ Зөвхөн нэг-оролт/нэг-гаралттай удирдлагын бүтцүүдийг ашигладаг
- ▶ Дүрмүүд:
  1. “Энгийн блок схем” —ээс эхэл
  2. Багцлах дүрэм: ямар ч тэгш өнцөгтийг (үйл) дараалсан хоёр тэгш өнцөгтөөр (үйлүүд) сольж болно
  3. Үүрлүүлэх дүрэм: ямар ч тэгш өнцөгтийг дурын удирдлагын бүтцээр (дараалал, **if**, **if...else**, **switch**, **while**, **do...while**, **for**) сольж болно
  4. Дүрэм 2, 3 ямарч дараалалтай, хэдэн ч удаа хэрэглэгдэж болно



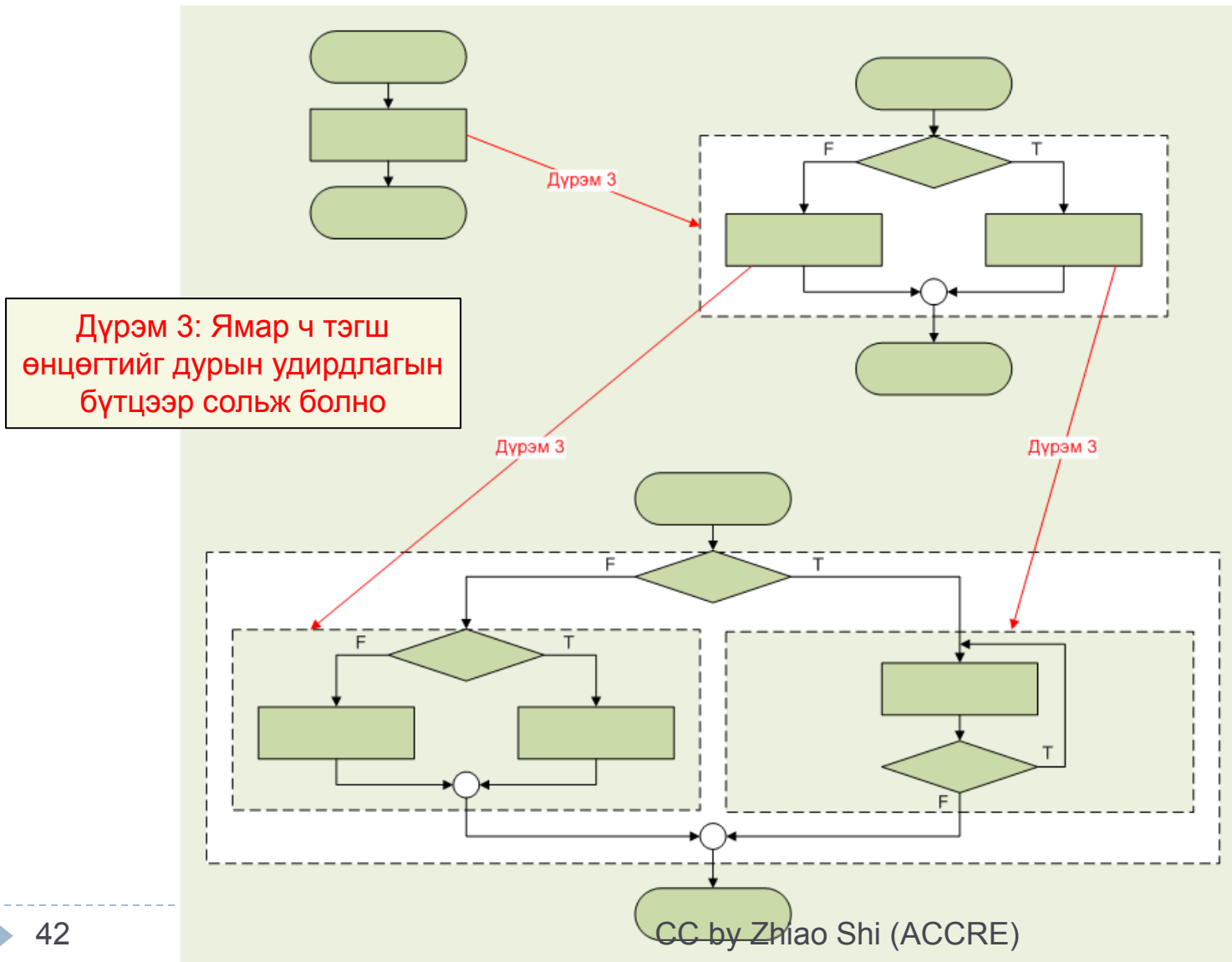
# Бүтцээр програмчлах асуудалд

Дүрэм 1: “Энгийн блок  
схем” –ээс эхэл

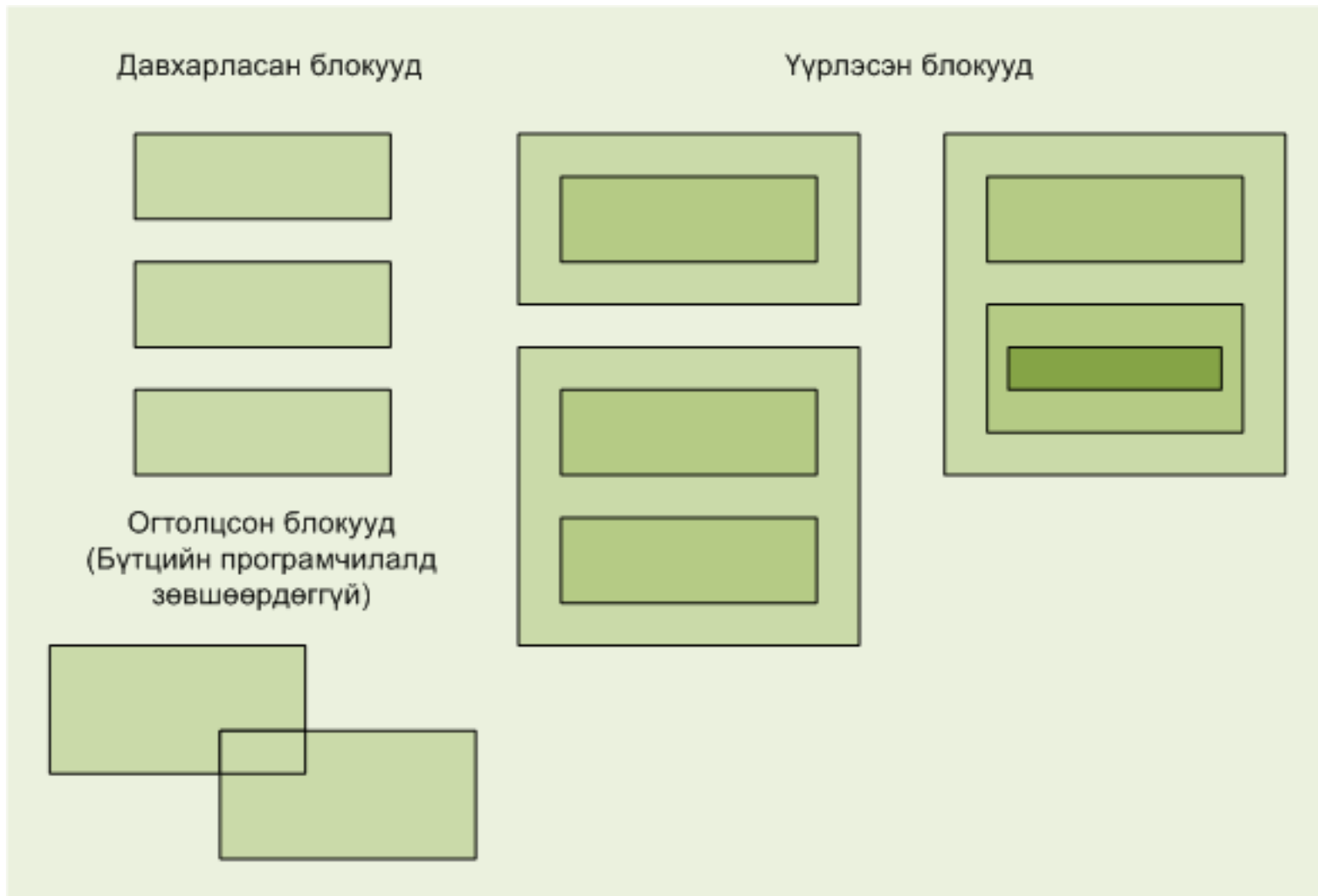
Дүрэм 2: Ямар ч тэгш өнцөгтийг  
хоёр тэгш өнцөгтөөр сольж  
болно



# Бүтцээр програмчлах асуудалд

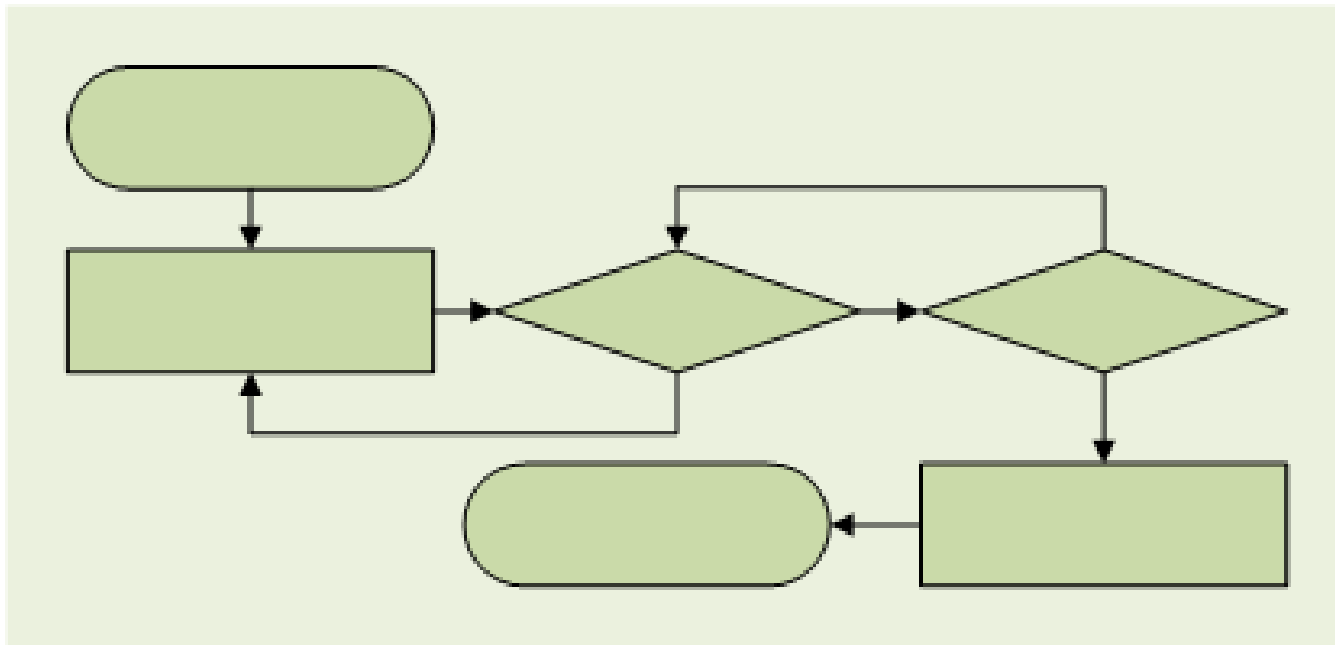


# Давхарласан, үүрлэсэн, огтолцсон бүтцийн блокууд



# Бүтэцлэгдээгүй блок схем

---



# Бүтцээр програмчлах асуудалд

---

- ▶ Ямар ч програмыг 3 удирдлагаар задалж болдог
  - ▶ Дараалал – хөрвүүлэгч автоматаар боловсуулдаг
  - ▶ Сонголт – `if`, `if...else`, `switch`
  - ▶ Давталт – `while`, `do...while`, `for`
    - ▶ Эдгээрийг зөвхөн дараах хоёр замаар нэгтгэж болно:
      - Үүрлүүлэх (дүрэм 3)
      - Багцлах (дүрэм 2)
- ▶ Ямар ч сонголтыг `if` оператороор, ямар ч давталтыг `while` оператороор бичиж болно

# Лекцийн агуулга

---

- ▶ **while, for, do...while** давталтын операторууд
- ▶ Програмын удирдлагын урсгалыг өөрчлөгч **break, continue** операторууд
- ▶ **switch** сонголтын оператор
- ▶ Логик үйлдэл ба удирдлагын операторын нийлмэл нөхцөл
- ▶ Тэнцэтгэл болон олгох үйлдлүүд
- ▶ Бүтцээр програмчлах асуудалд
- ▶ Дүгнэлт

# Дүгнэлт

- ▶ Давталтыг тоолууран болон онцгой утгаар удирдагдаж болно
- ▶ **for** давталтын оператор нь тоолууран удирдлагатай давталтын бүх ажиллагааг зохицуулдаг
- ▶ Ихэнхи тохиолдолд **for** давталтын операторыг түүний эквивалент болох **while** давталтаар орлуулж болно
- ▶ Давталт нь идэвхижүүлэлт, давталт төгсөх нөхцөл, нэмэгдүүлэх (хорогдуулах) гэсэн үе шаттай байдаг
- ▶ **double** төрлийн хөвөгч таслалтай тооны төрөл нь **float** төрөлтэй маш төстэй, ганцхан нарийвлал өндөртэй
- ▶ Хэвшүүлэгч мөр “%21.2f” нь хөвөгч таслалтай утгыг яг 21 тэмдэгтээр харуулахдаа таслалаас хойш 2 цифр байна гэдгийг зааж байна

# Дүгнэлт

---

- ▶ Удирдлагын бүтцийн дараалсан гүйцэтгэлийг өөрчлөхөд **break**, **continue** операторыг ашигладаг
- ▶ Олон сонголтыг **switch** оператороор хийж болно
- ▶ **do...while** бол давталтын өөр нэг оператор
- ▶ Логик **&&**(AND), **||**(OR), **!**(NOT) үйлдлүүд нийлмэл нөхцлийг үүсгэдэг
- ▶ Тэнцэтгэл (**==**) , олгох (**=**) үйлдлүүдийг будилж хэрэглэснээс ноцтой, илрүүлэхэд төвөгтэй алдаа гарч болзошгүй