

CS200 – Програмчлалын үндэс

Лекц 02

Бүтцээр Си програм хөгжүүлэх

Профессор А.Эрдэнэбаатар

Лекцийн агуулга

- ▶ Алгоритмын зохиомж
 - ▶ Блок схем ба псевдо код
- ▶ Удирдлагын бүтцүүд
 - ▶ Дарааллын бүтэц
 - ▶ Сонголтын бүтэц (**if** сонголтын оператор, **if...else** сонголтын оператор)
 - ▶ Давталтын бүтэц (**while** давталтын оператор)
- ▶ Дээрээс-Доош сайжруулалттай бүтцийн програмчилал
- ▶ Утга олгох, нэмэгдүүлэх, хорогдуулах үйлдэл
- ▶ Дүгнэлт

Алгоритм

- ▶ Алгоритмын зорилго бол Тооцооллын асуудал
 - ▶ Тодорхой дарааллаар бүлэг ажиллагааг явуулж асуудлыг шийдэж чадна.
- ▶ Алгоритм бол дараах ухагдахуунуудын цогц юм:
 - ▶ Биелэгдэх **ажиллагаанууд**
 - ▶ Эдгээр ажиллагаануудын **дараалал**
- ▶ Програмын удирдлага
 - ▶ Операторуудын биелэгдэх дарааллыг тодорхойлох
- ▶ Алгоритмыг дүрслэх хоёр хэлбэр
 - ▶ Псевдо код – өдөр тутмын хэлтэй төстэй
 - ▶ Блок схем – график дүрслэл

Псевдо код

▶ Псевдо код

- ▶ Алгоритмыг хөгжүүлэхэд бидэнд тусалдаг албан бус, зохиомол хэл
- ▶ Ердийн хэлтэй төстэй
- ▶ Компьютер дээр шууд ажиллахгүй
- ▶ Програмыг бичиж эхлэхийн өмнө “сэтгэн бодоход” бидэнд тусалдаг.
 - ▶ Си програм болгон хөрвүүлэхэд амар
 - ▶ Тайлбар, хувьсагчийн зарлагчгүй, зөвхөн биелэгдэх ажиллагааг агуулдаг

Блок схем

▶ Блок схем

- ▶ Алгоритмын график дүрслэл
- ▶ Тодорхой тооны тусгай зориулалт бүхий дүрсийг урсгал гэж нэрлэдэг сумаар холбож зурдаг
- ▶ Тэгш өнцөгт (ажиллагааны дүрс).
 - ▶ Аливаа төрлийн ажиллагааг илэрхийлнэ.
- ▶ Эллипс дүрс
 - ▶ Програм буюу түүний нэг хэсгийн эхлэл, төгсгөлийг заана.
- ▶ Удирдлагын бүтцийн нэг орц/нэг гарц
 - ▶ Нэг удирдлагын бүтцийн гарах цэг нөгөөгийн орох цэгтэй холбоддог (удирдлагын бүтцүүдийг залгах)
 - ▶ Програм бүтээлтийг хялбар болгодог

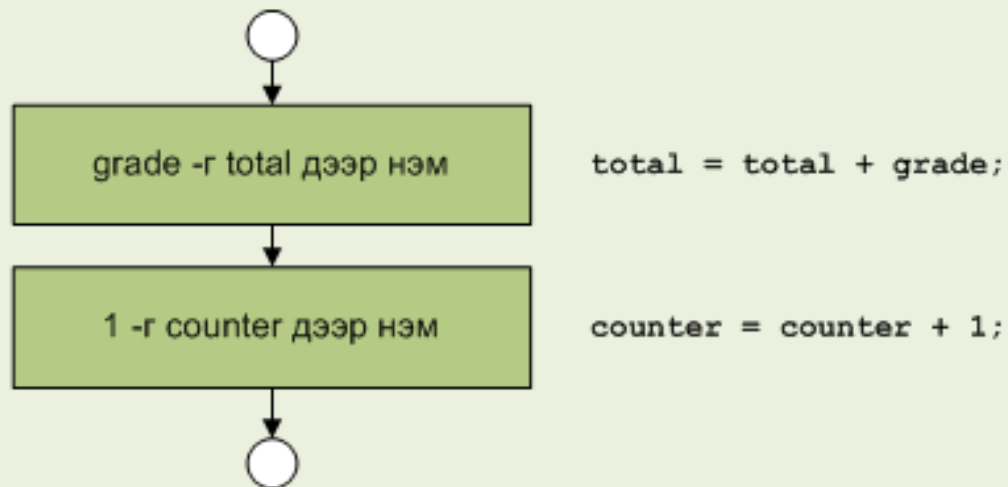
Лекцийн агуулга

- ▶ Алгоритмын зохиомж
 - ▶ Блок схем ба псевдо код
- ▶ Удирдлагын бүтцүүд
 - ▶ Дарааллын бүтэц
 - ▶ Сонголтын бүтэц (**if** сонголтын оператор, **if...else** сонголтын оператор)
 - ▶ Давталтын бүтэц (**while** давталтын оператор)
- ▶ Дээрээс-Доош сайжруулалттай бүтцийн програмчилал
- ▶ Утга олгох, нэмэгдүүлэх, хорогдуулах үйлдэл
- ▶ Дүгнэлт

Удирдлагын бүтцүүд

- ▶ Дараалан биелэгдэнэ (өгөгдмөл шинж)
 - ▶ Операторууд бичигдсэн дарааллаараа биелэгдэнэ
- ▶ Удирдлага дамжуулах
 - ▶ Дараачийн биелэгдэх оператор бичигдсэн дарааллаар биш байх
 - ▶ `goto` операторыг хэтрүүлж хэрэглэх нь асуудлыг төвөгтэй болгодог
- ▶ Бүх програм дараах 3 удирдлагын бүтцээр бичигддэг
 - ▶ **Дарааллын бүтэц:** Си хэлний үндсэн бүтэц. Өгөгдмөл шинжээр дараалан биелэгдэнэ.
 - ▶ **Сонголтын бүтэц:** 3 төрөл: `if`, `if...else`, `switch`
 - ▶ **Давталтын бүтэц:** 3 төрөл: `while`, `do...while`, `for`

Дарааллын бүтцийн блок схем



if сонголтын оператор

- ▶ Сонголтын бүтэц:

- ▶ Ажиллагааны альтернатив замналыг сонгоход ашигладаг

- ▶ Псевдо код:

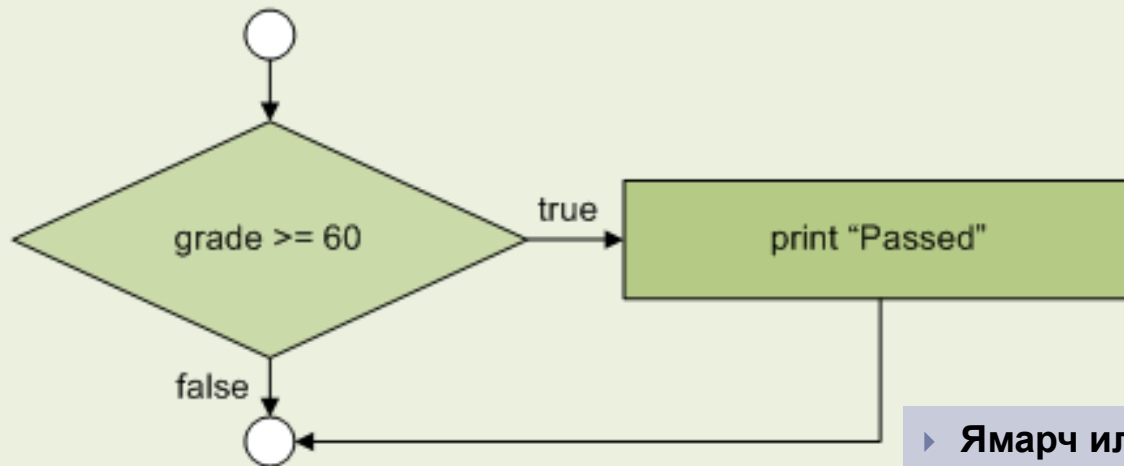
Хэрвээ оюутны үнэлгээ 60 –тай тэнцүү буюу их бол
“Тэнцсэн” гэдгийг хэвлэх

- ▶ **if** нөхцөл үнэн (**true**)

- ▶ Хэвлэх оператор хэрэгжиж, програм дараачийн операторт шилжинэ
- ▶ Хэрэв нөхцөл худал (**false**) бол хэвлэх операторыг алгасч програм дараачийн операторт шилжинэ
- ▶ Програмыг догол гаргаж бичих нь уншихад хялбар болдог.
 - ▶ Си хэлэнд хоосон зайн тэмтэгтийг алгасадаг

if сонголтын оператор

- ▶ Ромбо дүрс (шийдлийн дүрс)
 - ▶ Шийдэл гаргах ёстойг заадаг
 - ▶ Доторх илэрхийлэл нь true эсхүл false байна
 - ▶ Нөхцлийг шалгаад, зохих замыг сонгоно



- ▶ Ямарч илэрхийллээр шийд гаргаж болно
- ▶ 0 – false
- ▶ 0 биш – true
- ▶ Жишээ нь: 3 – 4 бол true

`if...else` сонголтын оператор

- ▶ **`if`**

- ▶ Зөвхөн нөхцөл `true` үед үйл хэрэгжинэ

- ▶ **`if...else`**

- ▶ Нөхцлийн үнэн, худлын тохиолдол бүрт **харгалзах** үйл хэрэгжинэ

- ▶ Псевдо код

Хэрвээ оюутны үнэлгээ 60 –тай тэнцүү буюу их бол

“Тэнцсэн” –г хэвлэ

эсрэг тохиолдолд

“Тэнцээгүй” –г хэвлэ

- ▶ Догол гаргаж бичсэнийг анхаар!

if...else сонголтын оператор

► Си код

```
if (grade >= 60 )  
    printf( "Passed\n" );  
else  
    printf( "Failed\n" );
```

► Гурван гишүүнт үйлдэл (? :)

- Гурван гишүүн авдаг (нөхцөл ? true утга : false утга)
- Өгөгдсөн псевдо кодыг ингэж бичнэ:

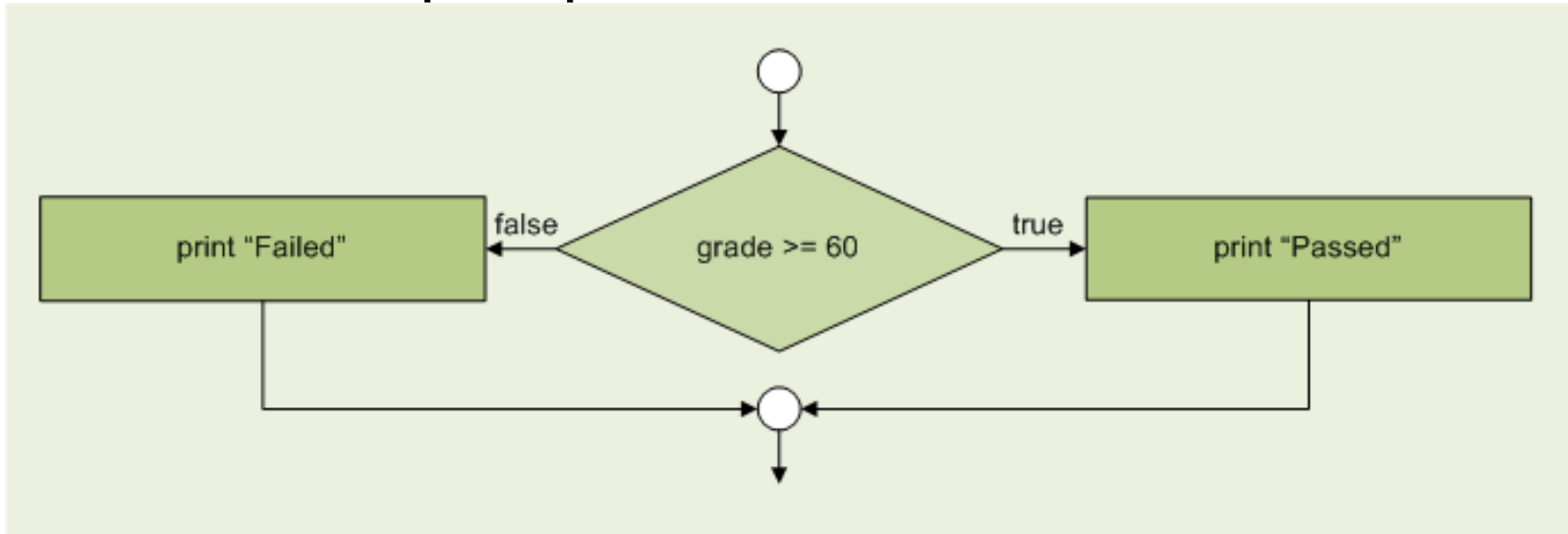
```
printf( "%s\n", grade >= 60 ? "Passed" : "Failed");
```

► Эсхүл:

```
grade >= 60 ? printf( "Passed\n" ) : printf( "Failed\n" );
```

if...else сонголтын оператор

▶ if...else операторын блок схем



▶ Үүрлэсэн if...else оператор

- ▶ if...else операторын дотор if...else –г давхар байршуулах замаар олон нөхцлийг шалгаж болно
- ▶ Нэгэнт нөхцөл биелсэн бол бусад операторуудыг алгасдаг
- ▶ Практикт хэт гүнзгий доголыг тэр бүр хэрэглэдэггүй

if...else сонголтын оператор

► Үүрлэсэн **if...else** операторын псевдо код

Хэрвээ оюутны үнэлгээ 90 –тай тэнцүү буюу их бол

“А” –г хэвлэ

эсрэг тохиолдолд

Хэрвээ оюутны үнэлгээ 80 –тай тэнцүү буюу их бол

“В” –г хэвлэ

эсрэг тохиолдолд

Хэрвээ оюутны үнэлгээ 70 –тай тэнцүү буюу их бол

“С” –г хэвлэ

эсрэг тохиолдолд

Хэрвээ оюутны үнэлгээ 60 –тай тэнцүү буюу их бол

“D” –г хэвлэ

эсрэг тохиолдолд

“F” –г хэвлэ

if...else сонголтын оператор

► Багц оператор

► Их хаалт дотор хэд хэдэн оператор орох

► Жишээ:

```
if (grade >= 60 )  
    printf( "Passed\n" );  
else {  
    printf( "Failed\n" );  
    printf( "You mast take this course again.\n" );  
}
```

► Багцыг илэрхийлэх их хаалтыг хэрэглээгүй бол
`printf("You mast take this course again.\n");`
ямар ч тохиолдод шууд биелэгдэх болно.

`if...else` сонголтын операторын алдаа

- ▶ Дүрмийн алдаа

- ▶ Багц операторын хаалтыг орхигдуулах
- ▶ Хөрвүүлэгч алдааг илрүүлдэг

- ▶ Логик алдаа

- ▶ Гүйцэтгэх хугацаанд нөлөөлдөг
- ▶ Энгийн алдаа: програм ажиллах боловч буруу хариу гардаг
- ▶ Ноцтой алдаа: програмаас шууд гардаг

while давталтын оператор

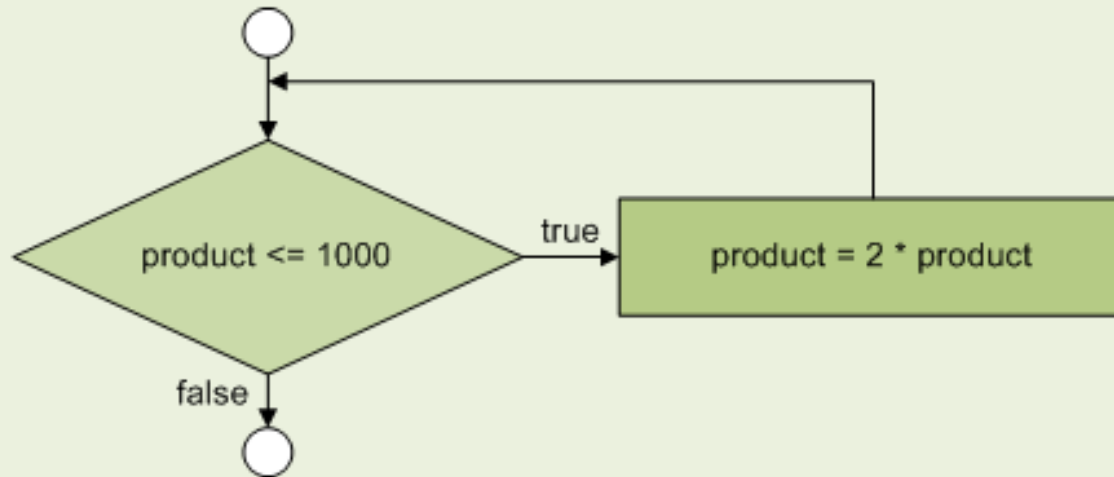
▶ Давталтын бүтэц

- ▶ Програм зохиогч ямар нэг нөхцлийг `true` байх тохиолдолд давтах үйлийг зааж өгдөг
- ▶ Псевдо код:
Миний худалдан авах юмны жагсаалт дуустал
Дараагийн юмыг авч, Жагсаалтнаас хас
- ▶ **while** давталт нөхцөл **false** болтол үргэлжилнэ

while давталтын оператор

► Жишээ:

```
int product = 2;  
while (product <= 1000 )  
    product = 2 * product;
```



while давталт операторын блок схем

Тоолууран удирдлагатай давталт

- ▶ Тоолуур тодорхой утга хүртэл давталт үргэлжилнэ
- ▶ Хязгаартай давталт: давталтын тоо тодорхой
- ▶ Жишээ: Ангийн 10 оюутан шалгалт өгчээ. Шалгалтын үнэлгээ (1 –с 100 хүртэлх оноо) өгөгдсөн бол ангийн дундаж үнэлгээг олъё
- ▶ Псевдо код:
 - Нийлбэр (total) тэнцүү 0
 - Үнэлгээний тоолуур (counter) тэнцүү 1
 - counter 10 –аас бага буюу тэнцүү бол давта
 - Дараачийн үнэлгээг (grade) оруул
 - grade –г total дээр нэм
 - counter дээр 1 –ийг нэм
 - Ангийн дундаж (average) total –ийг 10 –т хуваасантай тэнцүү
 - average –г хэвлэ

Давталтын жишээ

```
1.  /* Ex_06 Class average program with counter-controlled repetition */
2.  #include <stdio.h>
3.
4.  /* function main begins program execution */
5.  int main( void )
6.  {
7.      int counter; /* number of grade to be entered next */
8.      int grade;   /* grade value */
9.      int total;   /* sum of grades input by user */
10.     int average; /* average of grades */
11.
12.     /* initialization phase */
13.     total = 0;    /* initialize total */
14.     counter = 1; /* initialize loop counter */
15.
16.     /* processing phase */
17.     while ( counter <= 10 ) { /* loop 10 times */
18.         printf( "Enter grade: " ); /* prompt for input */
19.         scanf( "%d", &grade );    /* read grade from user */
20.         total = total + grade;     /* add grade to total */
21.         counter = counter + 1;     /* increment counter */
22.     } /* end while */
```

while давталтын тоолуур

Тоолуурын анхны утга 1

while давталт тоолуурын утга <= 10 байхад явна

Тоолуурыг нэмэгдүүлэх

Давталтын жишээ

```
23.  
24.    /* termination phase */  
25.    average = total / 10; /* integer division */ ← Дундажыг бодох  
26.  
27.    printf( "Class average is %d\n", average ); /* display result */  
28.  
29.    return 0; /* indicate program ended successfully */  
30. } /* end function main */  
Enter grade: 98  
Enter grade: 76  
Enter grade: 71  
Enter grade: 87  
Enter grade: 83  
Enter grade: 90  
Enter grade: 57  
Enter grade: 79  
Enter grade: 82  
Enter grade: 94  
Class average is 81
```

Лекцийн агуулга

- ▶ Алгоритмын зохиомж
 - ▶ Блок схем ба псевдо код
- ▶ Удирдлагын бүтцүүд
 - ▶ Дарааллын бүтэц
 - ▶ Сонголтын бүтэц (if сонголтын оператор, if...else сонголтын оператор)
 - ▶ Давталтын бүтэц (while давталтын оператор)
- ▶ Дээрээс-Доош сайжруулалттай бүтцийн програмчилал
- ▶ Утга олгох, нэмэгдүүлэх, хорогдуулах үйлдэл
- ▶ Дүгнэлт

Дээрээс-Доош, Алхам алхмаар сайжруулах

► Шийдвэрлэх асуудал

Ангийн дундажыг олох програмыг ажиллах бүртээ **дурын тооны** шалгалтын үнэлгээгээр дундажыг олдог байхаар хөгжүүлэх

- Оюутны тоо тодорхойгүй
- Програм хэзээ төгсөхөө яаж мэдэх вэ?

► Онцгой утга ашиглах

- Заримдаа дохио утга, тэнэг утга, дарцаг утга гэж нэрлэдэг
- Хэрэглэгч онцгой утга оруулахад давталт дуусна
- Онцгой утгыг сонгохдоо ердийн оруулах утгатай зөрчилгүй байлгахыг бод (энэ тохиолдолд -1 г.м.)

Дээрээс-Доош, Алхам алхмаар сайжруулах

- ▶ Дээрээс-доош, алхам алхмаар сайжруулах
 - ▶ Дээд псевдо кодын дүрслэлээс эхэл
Шалгалтын ангийн дундажыг бодох
 - ▶ Дээдээ жижиг бодлогод хувааж, дараалаар нь жагсаа
Хувьсагчдыг идэвхижүүл
Үнэлгээг оруулж, нийлбэрийг бодож, тоол
Ангийн дунджыг бодож, хэвлэ
- ▶ Ихэнх програм 3 үе шаттай байдаг
 - ▶ **Эхлэл**: програмын хувьсагчдыг идэвхижүүлэх
 - ▶ **Боловсруулалт**: өгөгдлийн утгыг оруулж, хувьсагчдыг өөрчлөх
 - ▶ **Төгсгөл**: тооцоолол, эцсийн үр дүнг хэвлэх

- ▶ Сайжруулалт бүр алгоритмыг төгс харуулах ёстой ба сайжралтууд нарийвчлалын түвшингээрээ ялгардаг.

Дээрээс-Доош, Алхам алхмаар сайжруулах

- ▶ *Хувьсагчдыг идэвхижүүл* үе шатыг сайжруулья
total 0 –тэй тэнцүү
counter 0 –тэй тэнцүү
- ▶ *Үнэлгээг оруулж, нийлбэрийг бодож, тоол* үе шатыг сайжруулья
Эхний үнэлгээг оруул (онцгой утга ч байж магадгүй)
Хэрэглэгч онцгой утга оруултал давта
grade –г явцын total дээр нэм
counter –г 1 –ээр нэмэгдүүл
Дараачийн үнэлгээг оруул (онцгой утга ч байж магадгүй)

Дээрээс-Доош, Алхам алхмаар сайжруулах

- ▶ *Ангийн дунджыг бодож, хэвлэ* үе шатыг сайжруулья

Хэрвээ counter 0 –тэй тэнцүү биш бол

average нь total –г counter –т хуваасантай тэнцүү

average –г хэвлэ

эсрэг тохиолдолд

“Үнэлгээ ороогүй байна” гэдгийг хэвлэ

Онцгой утган удирдлагатай давталт

```
1. total 0 -тэй тэнцүү
2. counter 0 -тэй тэнцүү
3.
4. Эхний үнэлгээг оруулья (онцгой утга ч байж магадгүй)
5. Хэрэглэгч онцгой утга оруултал давта
6.     grade -г явцын total дээр нэм
7.     counter -г 1 -ээр нэмэгдүүл
8.     Дараачийн үнэлгээг оруул (онцгой утга ч байж магадгүй)
9.
10. Хэрвээ counter 0 -тэй тэнцүү бол
11.     average нь total -г counter -т хуваасантай тэнцүү
12.     average -г хэвлэ
13. эсрэг тохиолдолд
14.     "Үнэлгээ ороогүй байна" гэдгийг хэвлэ
```

Дээрээс-Доош, Алхам алхмаар сайжруулалт хэзээ зогсох вэ?

- ▶ Дээрээс-доош, алхам алхмаар сайруулах процесс алгоритмын псевдо код хангалттай түвшинд нарийвчлагдсан ба псевдо кодыг Си хэлэнд хөрвүүлэхэд бэлэн болсон үед зогсоно.
- ▶ Ингэсний дараа Си програмыг хөгжүүлэх ажиллагаа хэвийн, шулуухан явагддаг.

Си програм

```
1.  /* Ex_07 Class average program with sentinel-controlled repetition */
2.  #include <stdio.h>
3.
4.  /* function main begins program execution */
5.  int main( void )
6.  {
7.      int counter; /* number of grades entered */
8.      int grade;   /* grade value */
9.      int total;   /* sum of grades */
10.
11.     float average; /* number with decimal point for average */
12.
13.     /* initialization phase */
14.     total = 0;     /* initialize total */
15.     counter = 0;   /* initialize loop counter */
16.
17.     /* processing phase */
18.     /* get first grade from user */
19.     printf( "Enter grade, -1 to end: " ); /* prompt for input */
20.     scanf( "%d", &grade );               /* read grade from user */
21.
```

float гэдэг төрөл нь
хувьсагч бүхэл тоо биш
гэдгийг заадаг

Си програм...

```
22. /* loop while sentinel value not yet read from user */
23. while ( grade != -1 ) {
24.     total = total + grade; /* add grade to total */
25.     counter = counter + 1; /* increment counter */
26.
27.     /* get next grade from user */
28.     printf( "Enter grade, -1 to end: " ); /* prompt for input */
29.     scanf("%d", &grade); /* read next grade */
30. } /* end while */

31. /* termination phase */
32. /* if user entered at least one grade */
33. if ( counter != 0 ) {
34.     /* calculate average of all grades entered */
35.     average = ( float ) total / counter; /* avoid truncation */
36.
37.     /* display average with two digits of precision */
38.     printf( "Class average is %.2f\n", average );
39. } /* end if */
40. else { /* if no grades were entered, output message */
41.     printf( "No grades were entered\n" );
42. } /* end else */

42. return 0; /* indicate program ended successfully */

43. } /* end function main */
```

While давталт хэрэглэгч
-1 гэсэн утга оруултал
үргэлжлэнэ

Хэрэглэгч ядаж 1 үнэлгээ
оруулсан уу?

total –г float төрөлд
хөрвүүлж бана

Хариуг таслалаас хойш
2 оронтой хэвлэнэ

Си програм

```
Enter grade, -1 to end: 75
Enter grade, -1 to end: 94
Enter grade, -1 to end: 97
Enter grade, -1 to end: 88
Enter grade, -1 to end: 70
Enter grade, -1 to end: 64
Enter grade, -1 to end: 83
Enter grade, -1 to end: 89
Enter grade, -1 to end: -1
Class average is 82.50
```

```
Enter grade, -1 to end: -1
No grades were entered
```


Үүрлэсэн удирдлагын бүтэц

▶ Асуудал

- ▶ Сургуулийн 10 оюутны шалгалтын үнэлгээ байна (1 – тэнцсэн, 2 - унасан)
- ▶ Үр дүнг шинжлэх програм бич. Хэрвээ 8 –аас илүү оюутан тэнцсэн бол “Raise tuition” гэдгийг хэвлэ

▶ Санамж

- ▶ Програм 10 үнэлгээг боловсруулах ёстой. Тоолууран удирдлагатай давталтыг ашиглана
- ▶ Хоёр тоолуур ашиглаж болно. Нэг нь тэнцсэн, нөгөө нь унасан оюутны тоо
- ▶ Үнэлгээ бүр 1 эсхүл 2 гэсэн тоо байна. Тоо 1 биш бол 2 гэж бодъё

Үүрлэсэн удирдлагын бүтэц

- ▶ Дээд түвшний төсөөлөл

Шалгалтын үр дүнг шинжлээд төлбөрийг нэмэх эсэхийг шийд

- ▶ Эхний сайжруулалт

Хувьсагчдыг идэвхижүүл

10 үнэлгээг оруулах ба тэнцсэн, унасаныг тоол

Шалгалтын товчоог хэвлэж, төлбөр нэмэх эсэхийг шийд

- ▶ *Хувьсагчдыг идэвхижүүлэл* -г сайжруулах

passes –г 0 болго

failures –г 0 болго

student тоолуурыг 1 болго

Үүрлэсэн удирдлагын бүтэц

- ▶ *10 үнэлгээг оруулах ба тэнцсэн, унасаныг тоол* -г сайжруулах

student тоолуур 10 –аас бага буюу тэнцүү бол давт

Дараачийн үнэлгээг оруул

Хэрэв оюутан тэнцсэн бол

passes 1 –ээр нэмэгдэнэ

эсрэг тохиолдолд

failures 1 –ээр нэмэгдэнэ

student тоолуур 1 –ээр нэмэгдэнэ

- ▶ *Шалгалтын товчоог хэвлэж, төлбөр нэмэх эсэхийг шийд* -г сайжруулах

passes –г хэвлэ

failures –г хэвлэ

Хэрэв 8 их оюутан тэнцсэн бол

“Raise tuition” –г хэвлэ

Үүрлэсэн удирдлагын бүтэц

```
1. passes -г 0 болго
2. failures -г 0 болго
3. student тоолуурыг 1 болго
4.
5. student тоолуур 10 -аас бага буюу тэнцүү бол давт
6.     Дараачийн үнэлгээг оруул
7.
8.     Хэрэв оюутан тэнцсэн бол
9.         passes 1 -ээр нэмэгдүүл
10.    эсрэг тохиолдолд
11.        failures 1 -ээр нэмэгдүүл
12.
13.    student тоолуур 1 -ээр нэмэгдүүл
14.
15. passes -г хэвлэ
16. failures -г хэвлэ
17. Хэрэв 8 их оюутан тэнцсэн бол
18.     "Raise tuition" -г хэвлэ
```

```

1.  /* Ex_08 Analysis of examination results */
2.  #include <stdio.h>
3.
4.  /* function main begins program execution */
5.  int main( void )
6.  {
7.      /* initialize variables in definitions */
8.      int passes = 0;    /* number of passes */
9.      int failures = 0; /* number of failures */
10.     int student = 1;   /* student counter */
11.     int result;        /* one exam result */
12.
13.     /* process 10 students using counter-controlled loop */
14.     while ( student <= 10 ) {
15.
16.         /* prompt user for input and obtain value from user */
17.         printf( "Enter result ( 1=pass,2=fail ): " );
18.         scanf( "%d", &result );
19.
20.         /* if result 1, increment passes */
21.         if ( result == 1 ) {
22.             passes = passes + 1;
23.         } /* end if */
24.         else { /* otherwise, increment failures */
25.             failures = failures + 1;
26.         } /* end else */
27.
28.         student = student + 1; /* increment student counter */
29.     } /* end while */

```

while давталт 10 оюутныг боловсруултал үргэлжилнэ

if болон else операторууд нь while давталтын дотор үүрлэж байна

Си програм...

```
30.
31.     /* termination phase; display number of passes and failures */
32.     printf( "Passed %d\n", passes );
33.     printf( "Failed %d\n", failures );
34.
35.     /* if more than eight students passed, print "raise tuition" */
36.     if ( passes > 8 ) {
37.         printf( "Raise tuition\n" );
38.     } /* end if */
39.
40.     return 0; /* indicate program ended successfully */
41.
42. } /* end function main */
Enter Result (1=pass, 2=fail): 1
Enter Result (1=pass, 2=fail): 2
Enter Result (1=pass, 2=fail): 2
Enter Result (1=pass, 2=fail): 1
Enter Result (1=pass, 2=fail): 1
Enter Result (1=pass, 2=fail): 1
Enter Result (1=pass, 2=fail): 2
Enter Result (1=pass, 2=fail): 1
Enter Result (1=pass, 2=fail): 1
Enter Result (1=pass, 2=fail): 2
Passed 6
Failed 4
```

Си програм...

```
Enter Result (1=pass, 2=fail): 1
Enter Result (1=pass, 2=fail): 1
Enter Result (1=pass, 2=fail): 1
Enter Result (1=pass, 2=fail): 2
Enter Result (1=pass, 2=fail): 1
Enter Result (1=pass, 2=fail): 1
Enter Result (1=pass, 2=fail): 1
Enter Result (1=pass, 2=fail): 1
Enter Result (1=pass, 2=fail): 1
Enter Result (1=pass, 2=fail): 1
Passed 9
Failed 1
Raise tuition
```

Лекцийн агуулга

- ▶ Алгоритмын зохиомж
 - ▶ Блок схем ба псевдо код
- ▶ Удирдлагын бүтцүүд
 - ▶ Дарааллын бүтэц
 - ▶ Сонголтын бүтэц (**if** сонголтын оператор, **if...else** сонголтын оператор)
 - ▶ Давталтын бүтэц (**while** давталтын оператор)
- ▶ Дээрээс-Доош сайжруулалттай бүтцийн програмчилал
- ▶ Утга олгох, нэмэгдүүлэх, хорогдуулах үйлдэл
- ▶ Дүгнэлт

Олгох үйлдлүүд

▶ Олгох үйлдлийн хураангуй бичлэг

`c = c + 3;`

гэдгийг нэмэх олгох үйлдлийн хураангуйгаар бичиж болно: `c += 3;`

▶ Операторын загвар

хувьсагч = хувьсагч үйлдэл илэрхийлэл;

энэ загварыг дараах байдлаар бас бичиж болно

хувьсагч үйлдэл = илэрхийлэл;

▶ Бусад олгох үйлдлийн жишээ:

`d -= 4` `(d = d - 4)`

`e *= 5` `(e = e * 5)`

`f /= 3` `(f = f / 3)`

`g %= 9` `(g = g % 9)`

Олгох үйлдлүүд

Олгох үйлдэл	Жишээ	Тайлбар	Олгох утга
<code>int c = 3, d = 5, e = 4, f = 6, g = 12;</code> байсан бол			
<code>+=</code>	<code>c += 7</code>	<code>c = c + 7</code>	<code>10 -> c</code>
<code>-=</code>	<code>d -= 4</code>	<code>d = d - 4</code>	<code>1 -> d</code>
<code>*=</code>	<code>e *= 5</code>	<code>e = e * 5</code>	<code>20 -> e</code>
<code>/=</code>	<code>f /= 3</code>	<code>f = f / 3</code>	<code>2 -> f</code>
<code>%=</code>	<code>g %= 9</code>	<code>g = g % 9</code>	<code>3 -> g</code>

Нэмэгдүүлэх, хорогдуулах үйлдэл

- ▶ Нэмэгдүүлэх үйлдэл ($++$)
 - ▶ $c += 1$ гэдгийн оронд хэрэглэж болно
- ▶ Хорогдуулах үйлдэл ($--$)
 - ▶ $c -= 1$ гэдгийн оронд хэрэглэж болно
- ▶ Эхлэн нэмэгдүүлэх, хорогдуулах
 - ▶ Хувьсагчийн өмнө ашиглана ($++c$, $--c$)
 - ▶ Илэрхийлэл бодогдохоос өмнө хувьсагч өөрчлөгдөнө
- ▶ Дараа нь нэмэгдүүлэх, хорогдуулах
 - ▶ Хувьсагчийн ард ашиглана ($c++$, $c--$)
 - ▶ Илэрхийлэл бодогдсоны дараа хувьсагч өөрчлөгдөнө

Нэмэгдүүлэх, хорогдуулах үйлдэл

- ▶ Хэрвээ $c = 5$ бол

```
printf( "%d", ++c);
```

- ▶ 6 –г ХЭВЛЭНЭ

```
printf( "%d", c++);
```

- ▶ 5 –г ХЭВЛЭНЭ

- ▶ Аль ч тохиолдолд c –ийн утга 6 болно

- ▶ Хувьсагч нь илэрхийлэлд ороогүй бол

- ▶ Эхлэн болон дараа нэмэгдүүлэх/хорогдуулахын хооронд ялгаа байхгүй

```
++c;
```

```
printf( "%d", c);
```

- ▶ Дараах жишээ адилхан үр дүн өгнө

```
c++;
```

```
printf( "%d", c);
```

Нэмэгдүүлэх, хорогдуулах үйлдэл

Үйлдэл	Жишээ илэрхийлэл	Тайлбар
++	++a	a –ийн утгыг 1 –ээр нэмэгдүүлнэ. Дараа нь шинэ утгыг хувьсагчийн орсон илэрхийлэлд ашиглана
++	a++	Хувьсагчийн тухайн утгыг хувьсагч орсон илэрхийлэлд ашиглана. Дараа нь a –ийн утгыг 1 –ээр нэмэгдүүлнэ.
--	--b	b –ийн утгыг 1 –ээр хорогдуулна. Дараа нь шинэ утгыг хувьсагчийн орсон илэрхийлэлд ашиглана
--	b--	Хувьсагчийн тухайн утгыг хувьсагч орсон илэрхийлэлд ашиглана. Дараа нь b –ийн утгыг 1 –ээр хорогдуулна.

```

1.  /* Ex_09 Preincrementing and postincrementing */
2.  #include <stdio.h>
3.
4.  /* function main begins program execution */
5.  int main( void )
6.  {
7.      int c;                /* define variable */
8.
9.      /* demonstrate postincrement */
10.     c = 5;                 /* assign 5 to c */
11.     printf( "%d\n", c );   /* print 5 */
12.     printf( "%d\n", c++ ); /* print 5 then postincrement */
13.     printf( "%d\n\n", c ); /* print 6 */
14.
15.     /* demonstrate preincrement */
16.     c = 5;                 /* assign 5 to c */
17.     printf( "%d\n", c );   /* print 5 */
18.     printf( "%d\n", ++c ); /* preincrement then print 6 */
19.     printf( "%d\n", c );   /* print 6 */
20.
21.     return 0; /* indicate program ended successfully */
22.
23. } /* end function main */
5
5
6
5
6
6

```

С –г хэвлээд дараа нь
нэмэгдүүлж байна

С –г нэмэгдүүлээд дараа
нь хэвлэж байна

Өмнө үзсэн үйлдлүүдийн ахлах чанар

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

Лекцийн агуулга

- ▶ Алгоритмын зохиомж
 - ▶ Блок схем ба псевдо код
- ▶ Удирдлагын бүтцүүд
 - ▶ Дарааллын бүтэц
 - ▶ Сонголтын бүтэц (if сонголтын оператор, if...else сонголтын оператор)
 - ▶ Давталтын бүтэц (while давталтын оператор)
- ▶ Дээрээс-Доош сайжруулалттай бүтцийн програмчилал
- ▶ Утга олгох, нэмэгдүүлэх, хорогдуулах үйлдэл
- ▶ Дүгнэлт

Дүгнэлт

- ▶ Псевдо код зөвхөн үйл заасан операторыг агуулдаг.
- ▶ Програмын псевдо кодын эхэнд бүх хувьсагч, тэдгээрийн зориулалтыг жагсаах нь зохимжтой.
- ▶ **Дарааллын, сонголтын, давталтын** гэсэн гурван төрлийн удирдлагын бүтэц байдаг.
- ▶ Блок схем бол алгоритмын график дүрслэл юм. Блок схемийг зурахдаа **тэгш өнцөгт**(үйл), **ромбо**(шийдэл), **эллипс**(оролт/гаралт), **жижиг тойрог**(холбоос) зэрэг тусгай зориулалтын дүрсүүдийг урсгал гэж нэрлэдэг **сумаар** (үйлийн дарааллыг харуулж) холбодог.
- ▶ **if, if...else, switch** гэсэн сонголтын операторууд байдаг.
- ▶ **while, do...while, for** гэсэн давталтын операторууд байдаг.

Дүгнэлт

- ▶ **if** операторын их биед хэд хэдэн операторуудыг оруулахын тулд их хаалтыг **{ }** хэрэглэдэг.
- ▶ Давталтыг тоолуурын болон онцгой утгын удирдлагатай зохион байгуулж болно.
- ▶ Дээрээс-доош, алхам алхмаар сайжруулах аргачлал нь сайн бүтэцлэгдсэн програмыг хөгжүүлэхэд маш чахал байдаг.
- ▶ Хөвөгч таслалтай тоог **float** төрлөөр илэрхийлдэг.
- ▶ Хөвөгч таслалтай тооцоолол хийхийн тулд бүхэл тоог хөвөгч таслал руу шилжүүлэх ёстой.
- ▶ Олгох үйлдлийг хураангуйлан илэрхийлж болно.
a = a + b; -> a += b;
- ▶ Нэг гишүүнтэй нэмэгдүүлэх **++**, хорогдуулах **--** үйлдэл байдаг.