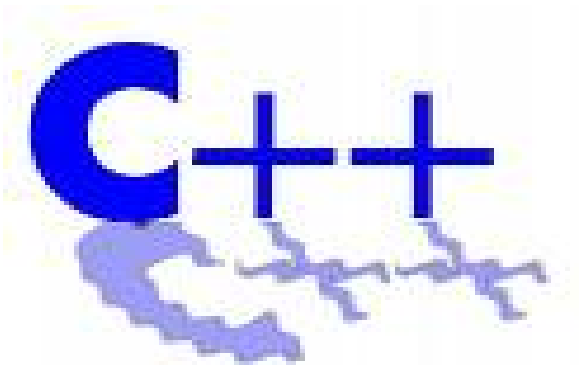


انٽروڊڪشن ٽو سي پلس پلس

خليق جمالي
0332-3803120
Sindh_desh@yahoo.com

وائيس آف سنڌِ اي بُک پبلشر

Join us on WhatsApp for daily newspapers and pdf books. #0344-3310224

فهرست

اڃنا

پنهنجي پاران

باب پهريون

سي پلس پلس ڇا آهي

باب ٻيو

سي پلس پلس پروگرامنگ بيسڪ

باب ٽيون

ويري ايبل

باب چوٿون

سي اين

باب پنجون

ارٿ ميٽڪ آپريٽرس

باب ڇهون

رليشن آپريٽرس

باب ستون

لوپس

باب اٺون

ڊسيشنس

باب نائون

فنڪشن

باب ڏهون

ورڪ وِٿ مُور

باب يارهون

آسڪي چارٽ

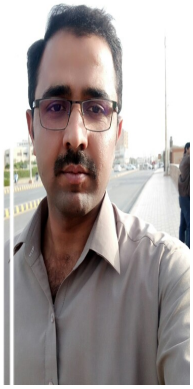
باب ٻارهون

آئوٽ پٽ سچاٽ

باب تيرهون

ايرر رپوٽنگ

Join us on WhatsApp for daily newspapers and pdf books. #0344-3310224



انفارميشن ٽيڪنالاجي جي نالي...

جنهن کان نہ صرف سامراج

پر موجوده دور ۾ قومي

تحريڪون پڻ

فائدو ونڊي

رهيون

آهن.

پنهنجي پاران:

جديد سائنس ڪمپيوٽر اڇڻ شرط ئي هن دنيا ۾ آئي آهي. ڪمپيوٽر جديد سائنس جو پيءَ آهي. اڄ ڪلهه دنيا ۾ ليٽسٽ ٽيڪنالاجي موبائيل فون آهي، جنهن کي پڙ اڪثر سائنسي ليڪڪ مينهي ڪمپيوٽر جي نالي سان سڏين ٿا. مطلب ته هاڻي سائنس جون جيڪي به ايجادون ٿيندیون سي ڪمپيوٽر ٽيڪنالاجي مان اخذ ڪيل هوندیون.

اسان جي سليبس ۾ هڪ عجيب فرضي سبق Age of probots ڏنل آهي، جنهن ۾ رائيٽر هڪ اهڙي پرسنل روبٽ جي ڳالهه ٿو ڪري جيڪو مستقبل ۾ انساني زندگي توڙي معاشري کي ڊرامائي انداز ۾ تبديل ڪري ڇڏيندو. انهي جو اثر شايد ڪمپيوٽر کان به وڌيڪ هجي، پر انهي سبق جي آخر ۾ رائيٽر ڇئي ٿو ته "پرسنل روبٽ دراصل پرسنل ڪمپيوٽر هوندو جنهن کي هٿ ۽ پير لڳل هوندا". معنيٰ ته مستقبل ۾ به ڪمپيوٽر جو استعمال ٿيندو.

انسان اڳي مشين کي هلائيندو هو پر ڪمپيوٽر جي نڪرڻ کان پوءِ پروگرامنگ ذريعي اسان مشينري کي پنهنجو غلام ڪري ورتو آهي، هاڻي انسان مشين کي حڪم ڏيندو آهي ۽ مشين طاقتور هوندي، نازڪ انسان سامهون جهڪي پوي ٿو. پروگرامنگ هڪ طريقي سان مشين ۽ انسان جي وچ ۾ هڪ پل جي حيثيت رکي ٿو.

ڪافي ماڻهو جديد توڙي قديم سائنس کي بغير ڪنهن مضبوط دليل جي مذهب سان ڳندين ٿا، ڪو چوي ٿو ته سائنس عيسائيت جي پيداوار آهي ته ڪو وري چوي ٿو ته سائنس اسلام جي پيداوار آهي. مطلب ته هر ڪو پنهنجي مذهب طرف سائنس کي ڇڪي ٿو جيڪو سراسر سائنس سان ناانصافي ڇئبي. ها! سائنس جي ترقي ۾ عيسائين به حصو ورتو ته مسلمانن به ورتو پر انهن جي مذهب سائنس لاءِ ڪجهه به ناهي ڪيو.

جي واقعي سائنس، اسلام ۽ عسيائيت مان اخذ ڪيل آهي ته پوءِ سِول انجنيئرنگ جو بي مثال ڪم موهن جو دڙو ڪيئن وجود ۾ آيو جڏهن ان وقت هنن مذهبن جا پيغمبر به پيدا نه ٿيا هئا. ڊارون جي نظريي کي هي مذهبي پيشوا چو نٿا مڃين، سائنس جي سڃي ڳالهه کي مذهب سان ٻڌي ان کي مذهبي بڻايو پيو وڃي، ۽ سائنس ۾ جيڪي شيون مذهب جي خلاف آهن وري انهن کي بدت ۽ الائي ڪهڙا نالا ڏئي رد ڪيو وڃي ٿو.

ڪي ماڻهو ته مرڳو سائنس جي مخالفت ڪري رهيا آهن انهن کي سمجهڻ گهرجي ته سائنس جي مخالفت انساني معاشري جي خودڪشي هوندي، ۽ جنهن ڏينهن سائنس هن دنيا مان پنهنجا پير اکوڙڻ شروع ڪيا ته پوءِ ان ڏينهن انسان به پنهنجا پير اکوڙڻ شروع ڪري.

ڪجهه ماڻهن جو خيال آهي ته شروعاتي دور ۾ سائنس نالي ڪا به شئي دنيا ۾ نه هئي پوءِ انسان ڪيئن ٿي زنده رهيو. معنيٰ ته سائنس جو هجڻ انسان لاءِ لازمي ناهي. جيڪڏهن هن دور ۾ سائنس ختم به ٿي وڃي ته پوءِ به انسان پنهنجي زندگي سکون سان گذاري سگهي ٿو. جيڪا ڳالهه پڙ انساني معاشري لاءِ فائدي واري ناهي، سائنس هن توڙي مستقبل جي معاشري لاءِ لازمي آهي، ان کي ختم نه ٿيڻ گهرجي.

جنهن سائنس اسان کي آسمانن جو سير ڪرايو، جنهن سائنس اسان کي پاڻي جي تهن تائين پهچايو، جنهن سائنس انسان جي ڪم ۾ مدد ڪرائي، جنهن سائنس مذهب جي گهڻن دعوائن کي غلط ڪري دنيا جي ماڻهن تان اهو ويساهه ختم ڪيو ته مذهب ئي آخري حل آهي انساني معاشري جي مسئلن جو. جنهن سائنس اسان کي گهڻ طرفي سوچڻ جو موقعو ڏنو، جنهن سائنس اسان کي گهڻو ڪجهه ڏنو چا اسان ان جا ٿورائتا ناهيون. پوءِ اسان اهڙي مهان سائنس جي مخالفت چو ڪري رهيا آهيون؟ چو اسان جو انسان اڃي سائنس جي مخالفت تي لٿو آهي،

ڪيس سمجهڻ گهرجي ته سائنس بنا دنيا تورا بورا جبل جو مثال ثابت ٿيندو. هن حسين دنيا کان سائنس نالي جياپي جي دوا ڪسي اوهان سک سان ويهندا؟. هرگز نه!.

اي سائنس جا مخالفو! صرف هڪ ڏينهن اوهان سائنس جي ايجادن کي استعمال نه ڪيو ۽ پوءِ ٻڌايو ته سائنس صحيح آهي يا اوهان...!

هتي منهنجو مطلب هرگز اهو ناهي ته ڪو آئون مذهبن يا ماڻهن تي تنقيد ڪريان. پر هي حقيقتون آهن جن کي لڪائڻ سان اهي وڌيڪ ڏپ ڪنديون. مون کي انهن ماڻهن سان چڙ آهي جيڪي سائنس جي بلاوجهه مخالف ٿيو بينا آهن. آئون دنيا جي اڪثر مسئلن جو حل سائنس کي سمجهندي ان جي طرف بيهان ٿو. دنيا وارا جهڙي طرف وڃن اُنهن سان مون کي رنج ناهي.

آخر ۾ آئون پنهنجي دوست پير فرحان احمد قريشي جا ٿورا مڃيندس. جيڪو هاسٽل جي مصروف زندگي مان ڏيلي ۾ ڪلاڪ ڪڍي مون کي سي پلس پلس متعلق ڪافي ڪجهه ٻڌائيندو هو.

خليق جمالي

28-07-2008

Join us on WhatsApp for daily newspapers and pdf books. #0344-3310224

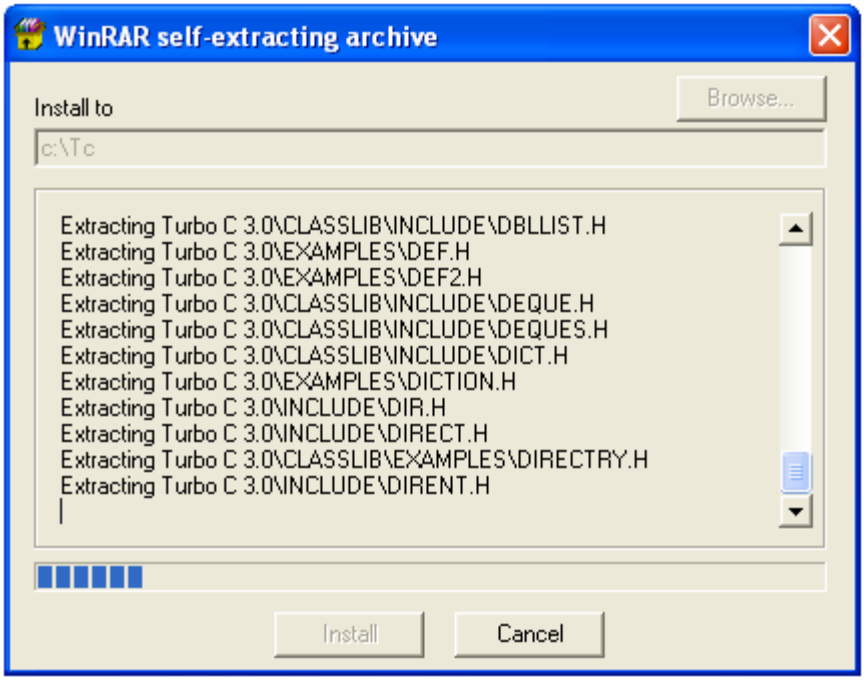
باب پهريون
سي پلس پلس ڇا آهي
What is c++

اسان عام زندگي ۾ بولي هڪ ٻئي کان ڄاڻ حاصل ڪرڻ لاءِ استعمال ڪندا آهيون. ساڳئي طرح،
سي پلس پلس (C++) هڪ بولي آهي جيڪا اسان استعمال ڪندا آهيون ڪمپيوٽر کي هدايت ڏيڻ لاءِ ۽ ڪمپيوٽر کان ڄاڻ حاصل ڪرڻ لاءِ. اوهان جيڪي سافٽويئر ڏسو ٿا، اهي پڻ هنن بولين جي پيداوار آهن.
هن وقت تائين گهڻيون ئي ڪمپيوٽر جي لاءِ بوليون ايجاد ڪيون ويون، انهن مان صرف ڪجهه ئي ڪامياب ٿي سگهيون، جن ۾ بيسڪ، فورٽران، ڪوبول، سي، سي پلس پلس، جاوا وغيره شامل آهن.
سي پلس پلس(C++) سي(C) مان اخذ ڪيل آهي.
ڪمپيوٽر جي هر بولي سان گڏ گهڻا ئي نوان فيچرس ايندا آهن. سي پلس پلس ۾ به گهڻا ئي نوان فيچرس آهن انهن ۾ سي پلس پلس جي ڊي اي (IDE) ، ڪمپائيلر، نوان ڪي ورڊ ۽ ٻيا به ڪوڙ سارا فيچرس.
ڪمپائير اسان جي لکيل ڪوڊنگ کي مشين جي بولي يعني 1 ۽ 0 ۾ ڪنورٽ ڪندو آهي.
آ ڊي اي (IDE Integrated development) هي اها جڳهه آهي جتي اوهان پنهنجي پروگرام لاءِ ڪوڊنگ لکي سگهو ٿا.

سي پلس پلس انسٽال ڪرڻ:
اوهان مارڪيٽ مان C++ TURBO 3.0 وڻي اچي پنهنجي ڪمپيوٽر تي استعمال ڪريو.
اوهان عام سافٽويئر وانگر سي پلس پلس کي انسٽال ڪري سگهو ٿا، جنهن جو هڪ مثال هيٺ ڏنل آهي.
سي پلس پلس مختلف سورس فائيلز ۾ اويليبل آهي.


Turbo C 3.0

اوهان ڪوشش ڪري سي پلس پلس هميشه C ڊرائيو ۾ انسٽال ڪريو.



سي پلس پلس شروع ڪرڻ:

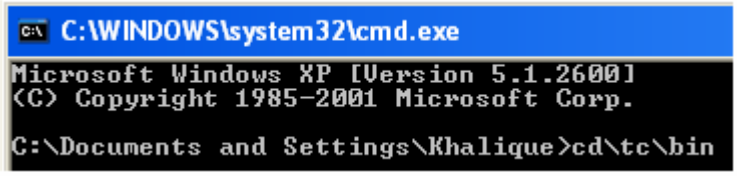
جڏهن اوهان سي پلس پلس پنهنجي ڪمپيوٽر ۾ انسٽال ڪندا ته ان ڊرائيو ۾ وڃي TC نالي سان سورس فائيل کي اوپن ڪريو.

مثال:

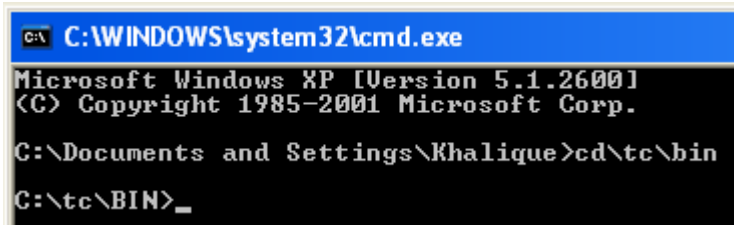
C => TC => BIN => TC.exe



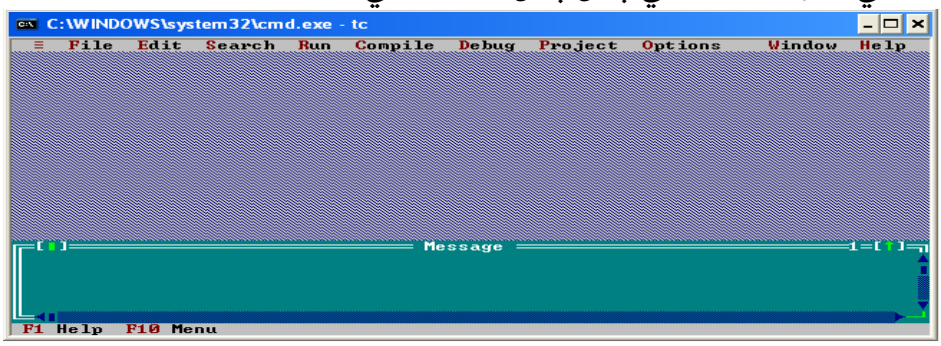
يا وري ڊاس ۾ وڃي هي ڪمانڊ لکي سي اوپن ڪريو. هي ڪمانڊ صرف ان وقت استعمال ڪريو جڏهن اوهان سي پلس پلس c ڊرائيو ۾ انسٽال ڪريو.



هاڻي انٽر ڪندا ته هينئان ڪمانڊ اچي ويندا.



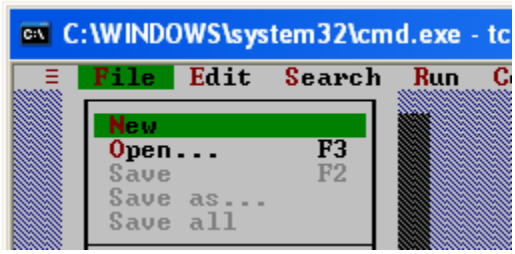
هاڻي اوهان هتي TC لکي انٽر ڪندا ته سي پلس پلس استارٽ ٿي ويندو.



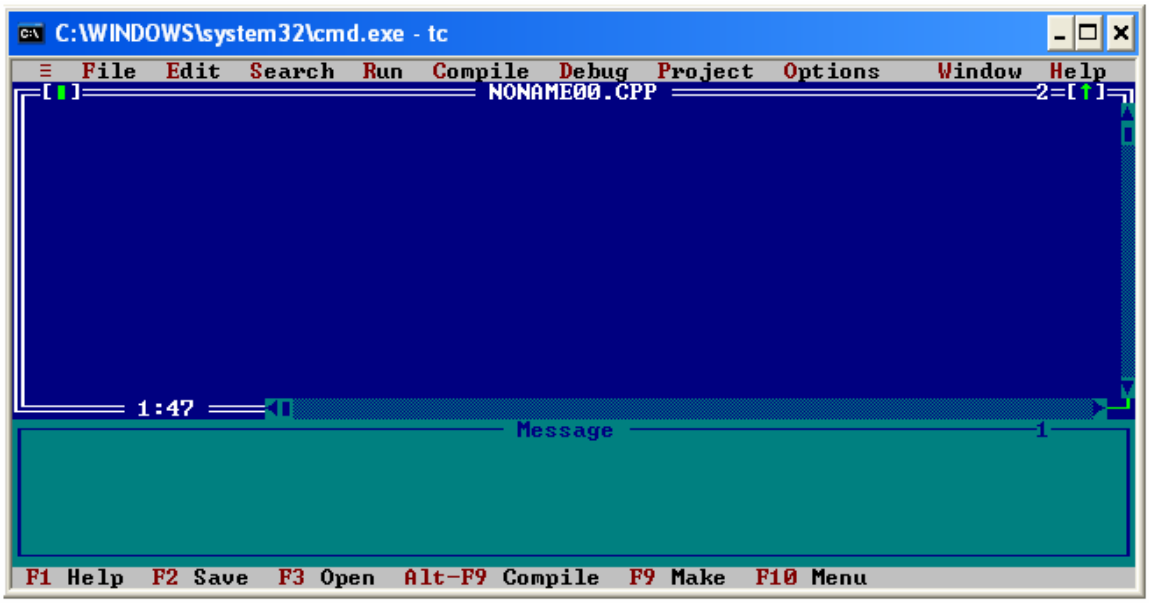
هي سي پلس پلس استارٽ ٿي ويو Alt + Enter دٻائي اسڪرين کي وڏو ڪريو.

سي ۾ ڪم شروع ڪرڻ:

جڏهن اوهان سي پلس پس استارٽ ڪري وڻندا ته اوهان عام ورڊ پروسسنگ سافٽويئر وانگر فائيل مينيو ۾ وڃي نيو New کي ڪلڪ ڪريو.



هاڻي اوهان وٽ سي پلس پلس جي آ ڊي اي کلي پوندي.



اوهانکي سي پلس پلس جي آ ڊي اي جي هيٺيان Message Box نظر اچي رهيو آهيو،اوهان جڏهن پروگرام ۾ غلط ڪوڊنگ استعمال ڪندا ته اهي هتي ظاهر ٿيندا.

جيڪڏهن اوهان سي پلس پلس کي C ڊرائيو ۾ انسٽال نٿا ڪريو ته اوهان آپشن مينيو ۾ وڃي ڊاريڪٽريز Directories کي ڪلڪ ڪريو.



۽ پوءِ اتان ڊاريڪٽريز کي C مان تبديل ڪري اهو ڊرائيو لکو جنهن ۾ سي پلس پلس انسٽال ڪيو آهي.



اوهان مٿي ڏسي رهيا آهيو ته ريڪمانڊيڊ ڊرائيو C آهي پر اوهان ان کي تبديل ڪري اها ڊرائيو لکو جنهن ۾ اوهان سي پلس پلس انسٽال ڪيو آهي، مثال اوهان E ڊرائيو ۾ انسٽال ڪيو آهي ته اوهان C کي تبديل ڪري E لکو.



ائين اوهان سي پلس پلس ۾ بغير ڪنهن ايرر جي ڪم ڪري سگهو ٿا، جي اوهان ائين نه ڪيو ته اوهان کي ايرر رپورٽ ٿيندو.

اوهان کي ڪوڊنگ جي باري ۾ ٻڌائڻ کان پهرين هڪ ڳالهه ٻڌائڻ لازمي سمجهان ٿو ته اوهان سي پلس پلس ۾ هيلپ Help کي استعمال ڪريو ته جيئن اجان وڌيڪ ڳالهيون سکي سگهو. اوهان کي سي پلس پلس جي هيلپ ۾ ڏنهيل پروگرام لاءِ ڪوڊنگ مهيا ڪئي وجي ٿي جن کي اوهان ڪاپي ڪري سي پلس پلس جي آ ڊي اي ۾ پيسٽ ڪري پروگرام هلائي سگهو ٿا.

باب ٻيو
سي پلس پلس پروگرامنگ بيسڪ

C++ Programming basic

اچو ته شروع ڪريون Geting started؛

اوهان ڪنهن به شيءِ جي باري ۾ بيسڪ ڄاڻ حاصل ڪرڻ کان سواءِ ان جا ماسٽر نٿا ٿي سگهو.

اهڙي طرح اوهان ڪمپيوٽر جي ٻولي جي باري ۾ شروعاتي شيون سکڻ کان سواءِ سنا پروگرامر نٿا ٿي سگهو.

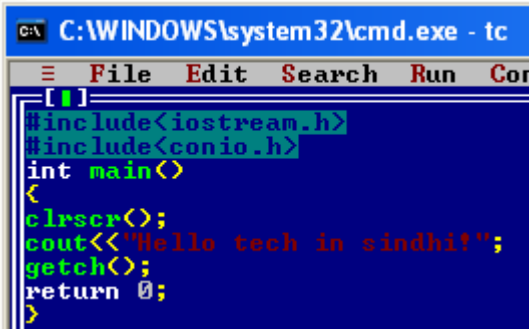
اچو ته شروعاتي پروگرامنگ سکون؛

اسين هڪ سادي پروگرام سان پنهنجي ٽيوٽر جي شروعات ڪنداسين. هن پروگرام جي ذريعي هڪ اسٽيٽمنٽ (هڪ سادو جملو) اسڪرين تي پرنٽ ڪنداسين.

پروگرام نمبر 1

```
#include<iostream.h>
#include<conio.h>
int main()
{
Clrscr();
cout<<"Hello tech in sindhi!";
getch();
return 0;
}
```

جڏهن هي ڪوڊنگ اوهان سي پلس پلس ۾ لکندا ته اهي هيٺيان نموني لکجي ايندا.



پروگرام رن Run (هلائڻ) ڪرڻ؛

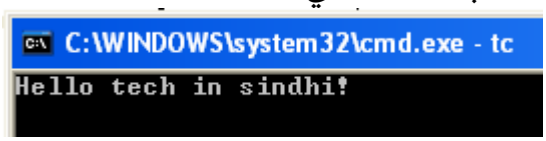
اوهان سي پلس پلس ۾ وڃي رن Run کي ڪلڪ ڪري مينيو مان رن کي ڪلڪ ڪريو.



يا وري اوهان ctrl+F9 پريس ڪريو.

آئوٽ پُٽ out put:

هن پروگرام ۾ هڪ استيٽمنٽ Hello tech in sindhi! اسڪرين ۾ پرنٽ ٿيندو. وري سي پلس پلس ۾ واپس اچڻ لاءِ انٽر Enter پريس ڪريو. اوهان آئوٽ پُٽ هيٺ ڏسي سگهو ٿا.



مٿين پروگرام جا فنڪشن:

هي سائيز جي حوالي سان هڪ ننڍو پروگرام آهي پر هن ۾ سي پلس پلس پروگرامنگ جي حوالي سان ڪافي بيسڪ شيون ڏنل آهن.

فنڪشن function:

سي پلس پلس ۾ فنڪشن پيڙهه واري پت جي حيثيت رکي ٿو، مٿين پروگرام ۾ ٽي فنڪشن استعمال ٿيل آهن. هڪ int main() , clrscr(); ۽ ٽيون getch();

Int main() کي اسان پروگرام جو گيٽ وي Gate way (داخلي رستو) ٻڌائي سگهون ٿا، int main() کان سواءِ اوهان جڏهن پروگرام رن ڪندا تہ اوهان وٽ ايرر رپورٽ ٿيندو.

clrscr(); هي Clear screen جو شارٽ فارم آهي، هي ڏاس ۾ جيڪو پراڻو پروگرام لڪيل هوندو ان کي مٽائي نئون پروگرام پرنٽ ڪرائيندو، اوهان هن سواءِ ٻ پروگرام ناهي سگهو ٿا پر جڏهن اوهان پنج وارا پروگرام کي رن Run ڪرائيندا تہ جيڪي پهريان چار پروگرام اوهان رن ڪرائيندا تہ پنجون ٻ انهن سان ملي ويندو. clrscr(); فنڪشن کي وڌيڪ سمجهڻ لاءِ مٿان پروگرام ٻ چڪر رن ڪريو هڪ چڪر clrscr(); سان گڏ ۽ ٻئي چڪر clrscr(); ڪٽي پوءِ رن ڪري ڏسو.

getch(); فنڪشن (هن کي گيٽ سي ايچ ڪري پڙهبو) get a character جو شارٽ فارم آهن، هن جو ڪم آهي تہ پروگرام کي اسڪرين تي بيهاري تہ جيئن اسان اُن کي آساني سان ڏسي سگهون.

فنڪشن نيم Function name:

هميشہ فنڪشن جي سڃاڻپ انهي جي پاسي کان ڏنل ٻہ بريسز ( ) آهن. جيڪڏهن اسان فنڪشن سان گڏ بريسز استعمال نہ ڪنداسين تہ ڪمپائيلر سمجهندو تہ هتي ڪو ويري ايبل (ويري ايبل جي متعلق اسان اڳتي سکنداسين) ڊڪليئر ڪيل آهي. ۽ ان جو نالو main رکيل آهي.

بريسز ۽ فنڪشن بادي Braces and function body:

جڏهن اوهان پروگرام ۾ ڪو فنڪشن استعمال ڪندا تہ ان جي شروعات ۾ { ۽ آخر ۾ } بريڪٽ ڏيڻ لازمي آهي، نہ تہ ٻئي صورت ۾ اوهان جو پروگرام ايرر رپورٽ ڪندو.

پروگرام اسٽيٽمنٽ Programm statement:

مٿين پروگرام ۾ ٻہ اسٽيمنٽ استعمال ٿيل آهن،
هڪ cout<<"Hello tech in sindhi!"; ۽ ٻيو return o;.

Retrun o; معنيٰ تہ هن پروگرام ۾ ڪو بہ ريٽرن وليليو استعمال ٿيل ناهي.

cout<<"Hello tech in sindhi!; هن جي ذريعي اسان جو لکيل اسٽيٽمنٽ اسڪرين تي پرنٽ ٿيندو.
هن جي لائين جي آخر ۾ سيمي ڪالن ; استعمال ٿيل آهي جنهن جو مطلب آهي تہ هتي لائين ختم ٿي وئي.
بيسڪ جهڙن ٻولين ۾ لائين بغير ڪنهن سيمي ڪالن جي ختم ڪري ڇڏيندا آهيون پر سي پلس پلس ۾ ائين ڪرڻ صحيح ناهي انهي لاءِ اوهان هر اسٽيمنٽ ۽ فنڪشن کان پوءِ (سواءِ main() ۽ ڪجهہ ٻين جي) سيمي ڪالن استعمال ڪندا ڪريو، نہ تہ اوهان جي پروگرام ۾ ايرر اچي سگهي ٿو.

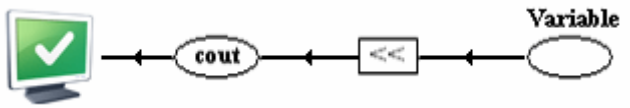
وائيٽ اسپيسز White spaces:

اوهان ڪنهن بہ پروگرام ۾ جيڪي خالي جايون ڇڏيو ٿا، جيئن اوهان ٻن اکرن جي وچ ۾ اسپيس بار استعمال ڪندا آهيو يا وري ٽيب (ٽيب اٺ ڪريڪٽر ڇڏيندو آهي) استعمال ڪندا آهيو، انهن کي ڪمپائيلر مڪمل طرح سان نظر انداز ڪري ڇڏيندو آهي.

آئوٽ پُٽ cout جو استعمال:

اوهان متي لکيل پروگرام تي نظر ڦيرائيندا ته اوهان کي cout<<"Hello tech in sindhi!" لکيل نظر ايندو. cout (هن کي سي آئوٽ ڪري اچاربوآهي) اهو پري ڊفائينڈ فنڪشن (اسان اڳتي هلي پري ڊفائينڈ فنڪشن جي باري ۾ سکنداسين) آهي، جيڪو اڳئي سي پلس پلس جي لائبريري ۾ محفوظ ٿيل آهي. cout هڪ آئوٽ پُٽ فنڪشن آهي ۽ اڪثر آئوٽ پُٽ فنڪشن اسڪرين تي ڪا به استيٽمنٽ پرنٽ ڪرائيندا آهن.

اوهان کي cout سان گڏ << (هن آپريٽر کي انسرشن آپريٽر ڪري اچاربو آهي) به نظر ايندو، هن جو ڪم آهي ته هن جي کابي پاسي جيڪا استيٽمنٽ يا اسٽرنگ ڪانسٽنٽ آهن ان کي cout ڏانهن ڪڍي ويندو ۽ cout ان استيٽمنٽ کي پرنٽ ڪرائيندو آهي.



فگر نمبر 1

مٿين فگر جي ذريعي اوهان سي آئوٽ ۽ انسرشن آپريٽر جو ڪم باآساني سمجهي سگهو ٿا.

انسرشن آپريٽر وري ايبل (استيٽمنٽ) کي سي آئوٽ ڏانهن موڪليندو آهن ۽ سي آئوٽ ان کي اسڪرين تي پرنٽ ڪرائيندو آهي.

اسٽرينگ ڪانسٽنٽ Straing constant:

اوهان کي استيٽمنٽ ۾ جيڪي به لفظ ڊبل ڪوٽيشن (" ") جي اندر لکيل نظر اچي رهيا آهن انهن کي اسٽرينگ ڪانسٽنٽ چئبو آهي.

```
"Hello tech in sindhi!"
```

توهان انگريزي لفظ ڪانسٽنٽ مان پلي پت واقف آهيو، جنهن جي معني آهي نه تبديل ٿيندڙ. ساڳي طرح سي پلس پلس ۾ به اسان جيڪا به شئي ڊبل ڪوٽيشن ۾ لکنداسين اها اسان جي اسٽرينگ ڪانسٽنٽ تي ويندي ۽ اها جيئن جو تيئن اسڪرين تي پرنٽ ٿيندي.

ڊريڪٽيوز Directives:

اوهان کي پروگرام نمبر 1 ۾ استعمال ٿيل پهريان به لائين نظر اچي رهيا آهن انهن کي ڊاريڪٽيوز جي نالي سان سڏجاتو وڃي ٿو.

ڊاريڪٽيوز سي پلس پلس جا حصا ته ناهن پر اسان کي هر صورت ۾ اهي استعمال ڪرڻا آهن.

پري پروسيسر ڊريڪٽيوز Preprocessor Directives:

```
#include<iostream.h>
#include<conio.h>
```

اوهان پروگرام نمبر 1 ڏسندا تہ اهو متين ڪوڊنگ سان شروع ٿئي ٿي. اوهان کي هڪ نشاني # نظر اچي رهي آهي، جنهن کي پري پروسيسر چئبو آهي.
#include ڪمپائيلر کي هدايت ڏيندو آهي تہ هڪ هيڊر فائيل کي هن پروگرام ۾ شامل ڪر، جيئن پروگرام سڌي نموني سان هلي سگهي.

هيڊر فائيل Header files:

```
<iostream.h>
<conio.h>
```

اوهان کي متي نظر ايندڙ ڪوڊنگ کي هيڊر فائيل چئبو آهي. دراصل هيڊر فائيل سي پلس پلس جي لائبريري ۾ اڳئي کان محفوظ ٿيل آهن، هن ۾ وڏي ڪوڊنگ استعمال ٿيل آهي، هيڊر فائيل پروگرام ۾ پنهنجن مخصوص فڪشن لاءِ ڪوڊنگ فراهم ڪندا آهن. جيئن iostream.h هيڊر فائيل <<cout لاءِ ڪوڊنگ فراهم ڪندي آهي ۽ conio.h هيڊر فائيل ;clrscr() ۽ ;getch() لاءِ ڪوڊنگ فراهم ڪندي آهي.
جيڪڏهن اوهان هيڊر فائيل جي ڪوڊنگ ڏسڻ چاهيو ٿا، تہ جيڪڏهن اوهان جو سي پلس پلس C ڊرائيو ۾ انسٽال ٿيل آهي تہ پوءِ

C:\tc\INCLUDE

اوهان اتي وڃي iostream نالي note paid فائيل کوليو اوهان ڏسندا تہ ان ۾ ڪيڏي نہ مشڪل ڪوڊنگ استعمال ٿيل آهي.
جيڪڏهن سي پلس پلس ۾ هيڊر فائيل ۽ پري پروسيسر ڊاريڪٽوز جا فيوچئرس نہ هجن ها تہ هيڊر فائيل ۾ استعمال ٿيل سڀ جو سڀ ڪوڊ اوهان کي هڪ پروگرام رن ڪرڻ لاءِ لکڻا پون ها.

ڪومنٽس Comments:

ڪومنٽس انگلش ٻولي جو لفظ آهي، جنهن جي معني آهي ڪنهن ماڻهو يا شئي جي باري ۾ تاثرات ڏيڻ.
ائين اسان جڏهن ڪنهن فنڪشن جي سامهون انهي بابت معلومات ڏيون ان کي سان ڪومنٽس چونداسين.
اوهان فرض ڪريو اوهان هڪ تمام وڏي ڪوڊنگ تي مشتمل پروگرام ناهيو ٿا، اوهان اهو پنج سال لاءِ پروگرام چڏيو ٿا، ۽ وري پنج سالن کان پوءِ ان تي نظر ڦيرايو ٿا، جيڪڏهن اوهان ڪومنٽس استعمال ڪندا تہ اوهان پنجن سان ڪانپوءِ بہ ان کي باآساني سمجهي سگهو ٿا. مطلب تہ ڪومنٽس پروگرام جي باري ۾ لکيل معلومات کي چئبو آهي.

ڪومنٽس جو استعمال Use of comments:

ڪومنٽس لکڻ جا ٻہ طريقا آهن هڪ ڪلاسڪ جيڪو پراڻي دور ۾ استعمال ٿيندو هو ۽ ٻيو نيو ٽائيپ ڪومنٽس جيڪي هيئنر استعمال ٿي رهيا آهن. ڪمپائيلر ڪومنٽس کي مڪمل طرح سان نظرانداز ڪري ڇڏيندو آهي. اسين پهرين ڪلاسڪ ڪومنٽس جو مثال ونڊاسين.

ڪلاسڪ ڪومنٽس Classic comments:

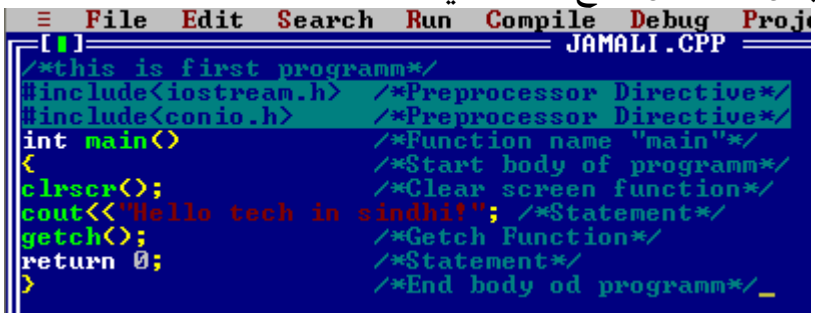
ڪلاسڪ ڪومنٽس هيئين طرح لکبا آهن.

/*This is old type comment*/

اسان پروگرام ۾ هنن جو استعمال هيئين طريقي سان ڪنداسين.

```
/*this is first programm*/
#include<iostream.h>  /*Preprocessor Directive*/
#include<conio.h>     /*Preprocessor Directive*/
int main()            /*Function name "main"*/
{                     /*Start body of programm*/
clrscr();             /*Clear screen function*/
cout<<"Hello tech in sindhi!"; /*Statement*/
getch();              /*Getch Function*/
return 0;             /*Statement*/
}                     /*End body old programm*/
```

هي ڪوڊنگ سي پلس پلس ۾ هيئين طرح نظر ايندي.



اوهان مٿي ڏسندا ته هر ڪنهن لائين جي سامهون ان جي باري ۾ معلومات ڏنل آهي.
هاڻي اچو ته ڪومنٽس کي ٻئي طريقي سان لکڻ سکون.

نيو ٽائيپ ڪومنٽس New type comments:

// this is new type comment

اچو ته ڏسون ڪيئن اسان هنن کي ڪيئن استعمال ڪنداسين.

```
//this is first programm
#include<iostream.h>  //Preprocessor Directive
#include<conio.h>     //Preprocessor Directive
int main()            //Function name "main"
```

```
{                       //Start body of programm
clrscr();               //Clear screen function
cout<<"Hello tech in sindhi!"; //Statement
getch();                //Getch Function
return 0;               //Statement
}                       //End body new programm
```

هاڻي اوهان باآساني ڪومنٽس کي استعمال ۾ آڻي سگهو ٿا. هي صرف معلومات فراهم ڪرڻ جو طريقو آهي، هن سان پروگرام ۾ ڪوبہ فرق نہ ايندو.

باب ٽيون
ويري ايبل
Variable

ويري ايبل ڪنهن به ڪمپيوٽر جي ٻولي جا اهم حصا آهن. دراصل جڏهن اسان ويري ايبل ڊڪليئر ڪندا آهيون، ته اها ڪمپيوٽر جي خاص ميموري ۾ هڪ جڳهه ناهيندي آهي، جنهن ۾ اسان پنهنجي ڊيٽا محفوظ ڪري سگهون ٿا. وري ايبل کي هڪ نالو ڏبو آهي، ۽ ان جي اندر مختلف قسم جي ڊيٽا رکي سگهجي ٿي. جڏهن اسان ويري ايبل ۾ ڪا به ڊيٽا رکندا آهيون اها ڪمپيوٽر جي خاص ميموري ۾ اسٽور ٿي ويندي آهي، جيڪا ويري ايبل پنهنجي لاءِ مخصوص ڪئي آهي.
انٽيجر، فلوٽ، ڪريڪٽر ۽ ٻين قسمن جا ويري ايبل سي پلس پلس ۾ ڪافي ڪارائتا آهن.

انٽيجر ويري ايبلInteger variable:

انٽيجر ويري ايبل ۾ اسان انگن واري ويليو محفوظ ڪري سگهون ٿا، مثال: 1،30،000 يا وري 27- وغيره.

ڊفائينگ انٽيجر ويري ايبلDefining integer variable:

انٽيجر ويري ايبل گهڻن ئي سائيز ۾ اسان ڊفائين ڪري سگهون ٿا. پر عام طور اسان int استعمال ڪندا آهيون. انٽيجر جي ميموري، سسٽم سان بدل آهي. اڪثر بٽ 32 (Bit 32) جي سسٽم ۾ انٽيجر چار بائيٽ (4 Bytes) والارينڊو آهي. انٽيجر ڊيٽا ٽائيپ ۾ اسان
-2,147,483,648 کان 2,147,483,647 تائين انگ محفوظ ڪري سگهون ٿا.
پر عام طور تي انٽيجر 4 بائيٽ جڳهه والارئيندو آهي، پر ايم ايس ڊاس ۽ موجوده آپريٽنگ سسٽم جهڙوڪ ونڊوز ۾ 2 بائيٽ ولارينڊو آهي.
اچو ته ڏسو پروگرام اندر ويري ايبل کي ڪيئن شامل ڪجي.

پروگرام نمبر 2

```
#include<iostream.h>
#include<conio.h>
int main()
{
clrscr();
int var1;       // Define var1
int var2;       // Define var2
getch();
return 0;
}
```

اوهان کي پروگرام نمبر 2 ۾ به نيون لائينون نظر اچي رهيون آهن، انهن مان هڪ int var1; ۽ ٻيو int var2; .

int جو ڪم آهي ڪمپيوٽر جي خاص ميموري ۾ هڪ مخصوص جڳهه ناهڻ جتي اسان پنهنجي انٽيجر ڊيٽا رکي سگهون. var1 ۽ var2 انهن مقرر ڪيل جڳهن جا نالا آهن. جيڪي انٽيجر مخصوص ڪيا آهن.
هتي int var1 ۽ int var2 کي ويري ايبل ڊيڪليئراشن (Declarations) چئبو آهي. اوهان هميشہ ويري ايبل کي ڊڪليئر ڪرڻ کان پوءِ ئي استعمال ڪري سگهو ٿا. وري ايبل کي آخر ۾ بين استيٽمنٽ جيان سيمي کالن ڏيڻ لازمي آهي.
اوهان پنهي ويري ايبلز کي هڪ لائين ۾ به ڊڪليئر ڪري سگهو ٿا. انهي لاءِ هينئين مثال ڏسو.

```
int var1, var2;
```

ويري ايبل نيمVariable name:

اوهان پروگرام نمبر 2 تي هڪ نظر ڦيرايو، اتي لکيل var1 ۽ var2 انهن ويري ايبلز جا نالا آهن جيڪي اسان ڊڪلئير ڪيا آهن. ۽ ڪنهن ڪمپيوٽر جي پولين ۾ هن کي ايڊنٽيفائر (identifiers) پڻ چيو وڃي ٿو.
اسان ويري ايبل جا نالا ڪهڙا مقرر ڪيون. پنهنجي جاءِ تي هي به هڪ اهم ڳالهه آهي. اسان A to Z يا 1 to 9 يا وري انڊراسڪور(_) به استعمال ڪري سگهون ٿا. هميشہ ويري ايبل جي نالي جي شروعات ۾ اکر يا انڊر اسڪور استعمال ٿيڻ گهرجي.
هميشہ ويري ايبل جو نالو يا ته مڪمل طرح ڪيپيٽل لفظن ۾ لکيل هجي يا ته مڪمل طرح سمال ليٽرز ۾ لکيل هجي. اسان ويري ايبل جي نالي جي شروعات ۾ انگ نٿا ڏئي سگهون، البته آخر ۾ ضرور استعمال ڪري سگهو ٿا.
اسان سي پلس پلس ۾ استعمال ٿيندر ڪي ورڊ (اسان ڪي ورڊ جي باري ۾ اڳتي سکنداسين) کي ويري ايبل جي نالي طور استعمال نٿا ڪري سگهون. هر اهو فڪشن جيڪو اسان سي پلس پلس ۾ لکون ۽ اهو اڇي رنگ جو ٿي وڃي ان کي ڪي ورڊ چئبو آهي، جيئن int وغيره.

اسائيمنٽ استيٽمنٽ Assignments statement:

اسائيمنٽ استيٽمنٽ (=) ساڄي پاسي واري ويليو کي کابي پاسي ويري ايبل ۾ سيو ڪندو آهي. اچو ته ڏسون اسائيمنٽ استيٽمنٽ سي پلس پلس ۾ ڪيئن لکجي.

```
int var1=20;

int var2=var1+10;
```

هاڻي مٿين استيٽمنٽ ۾ اسان هڪ ويري ايبل ڊڪليئر ڪيو آهي، جنهن جو نالو var1 آهي ۽ اسان ان جي ويليو 20 ڏني آهي. اسائيمنٽ آپريٽر 20 کي کٽي var1 ۾ سيو ڪندو.
ٻئي استيٽمنٽ ۾ اوهان کي + جي نشاني ڏسڻ ۾ اچي رهي آهي، جنهن کي ارٿ ميٽڪ آپريٽر پڻ چيو وڃي ٿو، سادي ميٿ جي قانون ۾ + ٻن عددن کي جوڙ ڪرڻ لاءِ استعمال ٿيندو آهي،ساڳي طرح سي پلس پلس ۾ + پڻ ٻن عددن يا وري ٻن ويري ايبل جي ويليوز کي جوڙ ڪرڻ لاءِ استعمال ٿيندو آهي.
var1 جنهن جي ويليو اسان پهرين استيٽمنٽ ۾ 20 مقرر ڪئي آهي، + آپريٽر var1 جي ويليو جيڪا 20 آهي ان ۾ 10 ٻيا جوڙ ڪندو ۽ اسائيمنٽ آپريٽر ٻنهي جي جوڙ يعني 30 کي var2 ۾ سيو ڪندو.
اهڙي طرح هاڻي اسان ويري ايبل کي هڪ مخصوص ويليو ڏيڻ ۾ ڪامياب ويا آهيون. اڳتي اسان ويري ايبل ۾ کي بورڊ ذريعي به ويليو ڏيڻ سکنداسين.

آئوٽ پٽ Out put:

هاڻي اوهان جڏهن هي پروگرام سي پلس پلس ۾ رن ڪرائيندا تہ ان لاءِ هيٺيان طرح ڪوڊنگ لکڻي پوندي. تہ جيئن اوهان پروگرام جو آئوٽ پٽ آساني سان ڏسي سگهو. دراصل اوهان مٿيان ٻہ لائين سي پلس پلس ۾ وڌيڪ لکڻيون پونديون، باقي پروگرام جي استيڪچر ساڳي ئي رهندي.

پروگرام نمبر 3

```
#include<iostream.h>
#include<conio.h>
int main()
{
clrscr();
int var1=20;
int var2=var1+10;
cout<<var2;
getch();
return 0;
}
```

پروگرام نمبر3 ۾ ويري ايبل جو ڪم اسان مٿي سکي آيا آهيون، اسان مٿي سکي آيا آهيون تہ cout جو ڪم آهي تہ ويليو يا اسٽرينگ ڪانسٽنٽ کي اسڪرين تي پرنٽ ڪرائڻ، cout<<var2; جو ڪم آهي تہ var2 جي ويليو کي اسڪرين تي پرنٽ ڪرائڻ، اسان کي خبر آهي تہ var2 جي ويليو 30 آهي ڇو تہ var1 جي ويليو اسان 20 رکي ۽ ان ۾ اسان 10 ٻيا جوڙ ڪيا انهن جو جواب 30 ٿيندو، ۽ اسائيمنٽ آپريٽر 30 کي var2 ۾ سيو ڪرائيندي، ۽ cout ، var2 جي ويليو يعني 30 کي اسڪرين تي پرنٽ ڪرائيندو.

اِين ڊل مينوپليٽر endl manipulator:

پروگرام نمبر4

```
#include<iostream.h>
#include<conio.h>
int main()
```

```
{
clrscr();
cout<<"Hello tech in sindh!"<<endl; //endl use for new line
cout<<"C++ in sindhi"<<endl;
cout<<"My computer & C++"<<endl;
getch();
return 0;
}
```

پروگرام نمبر 4 ۾ اوهان کي ٽي استيٽمنٽ نظر اچي رهيا آهن، اهي ٽئي اسٽرينگ ڪانسٽنٽ آهن .

اوهان کي پروگرام نمبر 4 ۾ هڪ نئون ۽ عجيب لفظ endl (اين ڊل) نظر اچي رهيو آهي، اوهان پهريان وجي پروگرام نمبر 1 تي نظر ڦيرايو، ان پروگرام ۾ اسان هڪ استيٽمنٽ Hello tech in sindhi ! اسڪرين تي پرنٽ ڪرائي، پر هن پروگرام ۾ اسان ٽي استيٽمنٽ هڪ جڳهه تي پرنٽ ڪرايا آهن.

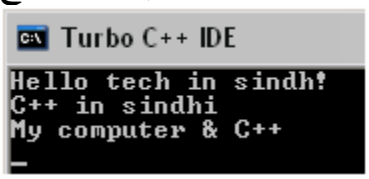
1. Hello tech in sindhi !

2. C++ in sindhi

3. My computer & C++

هتي سوال هي آهي ته اسان جڏهن پروگرام نمبر 1 ۾ اين ڊل استعمال نه ڪيو ته پوءِ اسان پروگرام نمبر 4 ۾ اين ڊل جو استعمال ڇو ڪري رهيا آهيون.

جڏهن اوهان هن پروگرام کي رن ڪرائيندا ته ان جو آئوٽ پُٽ هن طرح ايندو.



اسان سي پلس پلس ۾ اِين ڊل جي جڳهه تي \n به استعمال ڪندا آهيون.

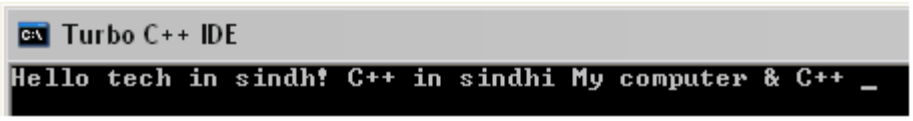
```
cout<<"Hello tech in sindh!\n";
```

پروگرام نمبر 5

```
#include<iostream.h>
#include<conio.h>
int main()
{
clrscr();
cout<<"Hello tech in sindh!";
cout<<"C++ in sindhi";
cout<<"My computer & C++";
getch();
return 0;
}
```

پروگرام نمبر 5 ۾ اسان اِين ڊل جو استعمال نه ڪيو آهي.

جڏهن اسان اين ڊل جو استعمال نه ڪيو ته ان پروگرام جو آئوٽ پٽ هن طرح آيو آهي.



هاڻي مون کي يقين آهي ته اوهان عجيب لفظ اين ڊل جي ڪم کان واقف ٿيا هوندا.
endl جو ڪم آهي ته هر اسٽيٽمنٽ کي نيئن لائين ۾ پرنٽ ڪرائڻ.
اوهان پروگرا نمبر 4 ۽ پروگرام نمبر 5 جي آئوٽ پُٽ کي ڏسي اين ڊل جي ڪم کان باآساني واقف ٿي ويا هوندا.

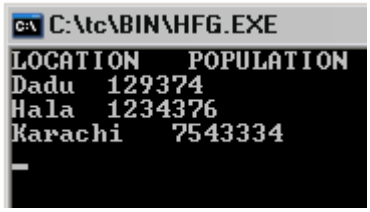
سيٽ ڊبليو setw :

سيٽ ڊبليو ، سيٽ ورٿ (Set width) جو شارٽ فارم آهي هن جو ڪم آهي ته ورٿ کي سيٽ ڪرڻ. اسان سيٽ ڊبليو پروگرام کي خوبصورت لوڪ Look ڏيڻ لاءِ استعمال ڪندا آهيون.
هتي هڪ پروگرام جو مثال ونو جيڪو بغير سيٽ ڊبليو جي نهيل آهي.

پروگرام نمبر 6

```
#include<constream.h>
int main()
{
clrscr();
cout<<"LOCATION   POPULATION"<<endl;
cout<<"Dadu  129374"<<endl;
cout<<"Hala  1234376"<<endl;
cout<<"Karachi   7543334"<<endl;
getch();
return 0;
}
```

پروگرام جو آئوٽ پٽ ڪجهه هن ريت نڪرندو.



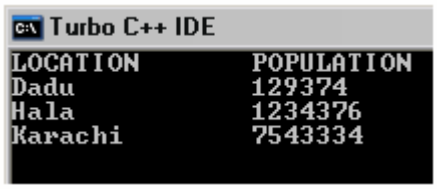
هاڻي اچو ته سيٽ ڊبليو جي مدد سان ناهيل هي پروگرام ڏسون.

پروگرام نمبر 7

```
#include<constream.h>
#include<iomanip.h>
int main()
{
clrscr();
cout<<setw(6)<<"LOCATION"<<setw(15)<<"POPULATION"<<endl;
cout<<setw(6)<<"Dadu"<<setw(15)<<"129374"<<endl;
```

```
cout<<setw(6)<<"Hala"<<setw(15)<<"1234376"<<endl;
cout<<setw(6)<<"Karachi"<<setw(15)<<"7543334"<<endl;
getch();
return 0;
}
```

هن پروگرام جو آئوٽ پٽ هن ريت نڪرندو.



هن پروگرام جي آئوٽ پُٽ کي سمجهڻ تمام سولو آهي، setw(6) مطلب تہ 6 ڪريڪٽر ڇڏي پوءِ هڪ استيٽمنٽ پرنٽ ڪرائي. Setw(15) مطلب آهي تہ 15 ڪريڪٽر ڇڏي پوءِ ٻئي استيٽمنٽ پرنٽ ڪرائي.
Iomanip هيڈر فائيل سيٽ ڊبليو لاءِ استعمال ٿيل آهي.

ڪريڪٽر ويري ايبلCharacter variable:

جيئن اسان مٿي سکي آيا آهيون تہ انٽيجر ويري ايبل ۾ اسان انگن واري ويليو سيو ڪري سگهون ٿا، ساڳي طرح اسان ڪريڪٽر ويري ايبلز آلفابيٽڪ ويليو ڏئي سگهون ٿا. جئين A, B, C, $, @ وغيره.
ڪريڪٽر ويري ايبل جي ويليو 0 کان 127 تائين آهي، پر ڪن آپريٽنگ سسٽم ۾ ڪريڪٽر جي ويليو 0 کان 256 تائين پڻ آهي.

ڪريڪٽر ويري ايبل ڊڪليئرDeclaration of ch:v:able:

ڪريڪٽر ويري ايبل ۽ انٽيجر ويري ايبل جي ڊڪليئر ڪرڻ واري سنٽيڪس Syntax ساڳي آهي. اسان انٽيجر ۾ int استعمال ڪندا آهيون ۽ ڪريڪٽر ۾ char استعمال ڪنداسون.

```
char a;
char b;
```

يا وري

```
char a,b,c,d,e;
```

پهرين اسان A ۽ B کي الڳ الڳ ڊڪليئر ڪيوسين پر تي نمبر استيٽمنٽ ۾ اسان A B C D E کي هڪ ئي جڳهہ تي ڊڪليئر ڪيو آهي.

ڪريڪٽر ڪانسٽنٽCharacter constant:

اسان ڪريڪٽر ڪانسٽنٽ کي سنگل ڪوٽيشن ۾ لکندا آهيون. مثال 'a' يا وري 'b' وغيره.

پروگرام نمبر 8

```
#include<iostream.h>
#include<conio.h>
int main()
{
clrscr();
char var1='A';
char var2='B';
cout<<var1;
cout<<var2;
getch();
return 0;
}
```

اوهان مٿين پروگرام ۾ ڪريڪٽر ڪانسٽنٽ جي ڪم کي باآساني سمجهي سگهو ٿا.
پهريان به ڪريڪٽر ويري ايبل ڊڪليئر ڪيل آهن هڪ جو نالو var1 ۽ ٻئي جو نالو var2 ڏنل آهي. var1 جي ويليو A ۽ var2 جي ويليو B مقرر ڪيل آهي.
cout ذريعي انهن ويري ايبلز جي ويليو پرنٽ ڪرايل آهي. هن پروگرام جو آئوٽ پٽ هن طرح ايندو.



فلوٽ ٽائيپ ويري ايبلFloat type variable؛

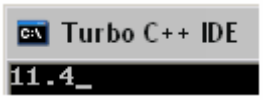
انٽيجر ويري ايبل ۾ اسان بغير ڪنهن ڏهائي عدد جي ويليو وڌي سگهون ٿا، پر فلوٽ ويري ايبل ۾ اسان ڏهائي واري عدد ۾ وڌي سگهون ٿا. مثال 1.50 يا وري 234.99 وغيره.

پروگرام نمبر 9

```
#include<iostream.h>
#include<conio.h>
int main()
{
clrscr();
float a=10.2;
float b;
b=a+1.2;
cout<<b;
getch();
return 0;
}
```

هي پروگرام به پروگرام نمبر 3 جيان آهي، پر هن ۾ صرف هڪ فرق آهي ته هن پروگرام ۾ ڏهائي استعمال ڪيل آهي، جڏهن ته پروگرام نمبر 3 ۾ ڏهائي استعمال ٿيل ناهي.
جيڪڏهن اوهان کي هي پروگرام سمجهڻ ۾ مشڪلات محسوس ٿي رهي آهي ته پوءِ پروگرام نمبر 3 جا انسٽرڪشن پڙهو.

هن پروگرام جو آئوٽ پُٽ هن طرح نڪرندو.



باب ڇوٿون
سي اين
Cin

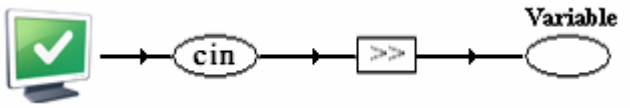
سي اين (cin) ڪون سول اِن پُٽ Consol input جو شارٽ فارم آهي. cin stand for consol in put
اسان سي اين جي ذريعي ڪي بورڊ مان ڪنهن ويري ايبل کي ويليو ڏئي سگهون ٿا.

سي اين سنٽيڪسCin syntex :

اچو ته هاڻي ڏسون ته سي اين کي سي پلس پلس ۾ ڪيئن لکجي.

```
cin>>a;
```

>> ايڪسٽريڪشن آپريٽر جو ڪم آهي ته اسان جيڪي به ويليو سي اين ذريعي ونڊون ان کي ويري ايبل a ۾ سيو ڪري.



فگر نمبر2

اوهان مٿين فگر ۾ ڏسي سگهو ٿا ته ڪي بورڊ مان ڏنل ويليو پهريان سي اِن ۾ داخل ٿيندي آهي، پوءِ ايڪسٽريڪشن آپريٽر ان کي ويري ايبل ۾ سيو ڪندو آهي.
اچو ته سي اين جو هڪ مثال ڏسون.

پروگرام نمبر 10

```
#include<constream.h>
int main()
{
clrscr();
int a,b;
cout<<"Enter 1st Value"<<endl;
cin>>a;
cout<<"Enter 2nd Value"<<endl;
cin>>b;
cout<<"Addition is "<<a+b;
getch();
return 0;
}
```

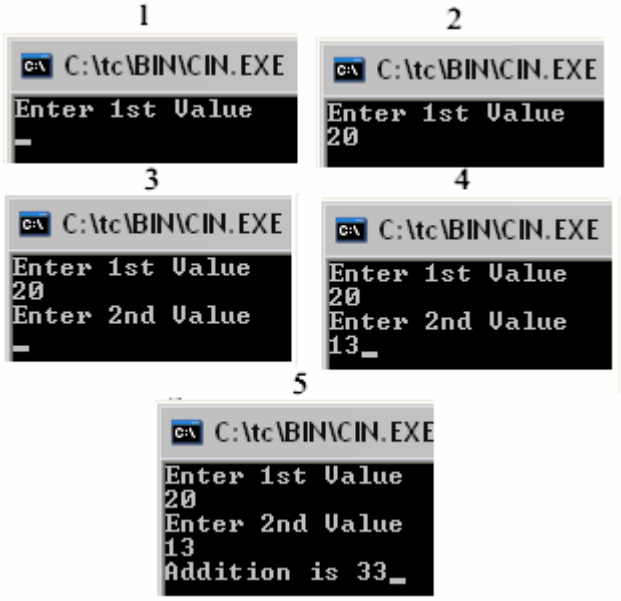
اوهان پروگرام نمبر 10 تي هڪ نظر وجهو، اوهان کي پهريان پهريان هڪ نئون هيڈر فائيل constream نظر اچي رهيو آهي.

ڪونسٽريم هيڈر فائيلconstream header file:

اسان ڪونسٽريم هيڈر فائيل iostream ۽ conio جي جڳهه تي استعمال ڪندا آهيون، جتي اسان اڳ ۾ هيڈر فائيل استعمال ڪندا هئاسين هاڻي اتي اسان هڪ هيڈر فائيل ڪونسٽريم استعمال ڪنداسين.

پروگرام نمبر 10 ۾ پهريان ته استرينگ ڪانسٽنٽ Enter 1st value پرنٽ ٿيندو، پوءِ اتي ڪرسر اچي ويندو، اوهان هاڻي ڪي بورڊ مان هڪ انٽيجر ويليو ڏيو مثال 20 ۽ پوءِ انٽر پريس ڪريو، جيئن اوهان انٽر پريس ڪندا ته وري ٻئي استرنگ ڪانسٽنٽ Enter 2nd value اسڪرين تي پرنٽ ٿيندو، ۽ وري هيٺ ڪرسر اچي ويندو، اوهان هاڻي ٻي ويليو ڏيو، مثال 13 ۽ پوءِ اوهان جيئن انٽر پريس ڪندا ته ٽيئن استرنگ ڪانسٽنٽ Addition is پرنٽ ٿيندو، ان جي سامهون ويري ايبل A ۽ وري ايبل B جو ويليو جوڙ ٿيل نظر ايندي، ويري ايبل A ۽ B جو جوڙ 33 پرنٽ ٿيندو.

اچو ته ڏسون سي پلس پلس ۾ هن پروگرام جو آئوٽ پٽ چا ٿو نڪري.



اسان سي اين جي ذريعي هڪ ڪان وڌيڪ ويري ايبلز جي ويليو هڪ ئي استيٽمنٽ ۾ ونيِ سگهون ٿا.

```
cin>>a>>b;
```

باب پنجون
ارٿ ميٽڪ آپريٽر

Arithmetic operators

سي پلس پلس ۾ پنج آرٿ ميٽڪ آپريٽر آهن، جيڪي سي پلس پلس ۾ به ميٿ وانگر جوڙ ۽ ڪٽ وغيره لاءِ استعمال ٿيندا آهن. اهي پنج ارٿ ميٽڪ آپريٽر هيٺ ڏجن ٿا.

آپريٽر	نالو
+	پلس
-	مائينس
*	ملٽيپليڪش
/	ڊيويزن
%	ريمائينڊر

اسان ارٿ ميٽڪ آپريٽر جو استعمال هر قسم جي ويري ايبل تي ڪري سگهون ٿا، انٽيجر ۽ فلوٽنگ ويري ايبل سان پڻ استمال ڪري سگهون ٿا. جيئن ٻين ٻولين ۾ ارٿ ميٽڪ جو استعمال آهي، سي پلس پلس ۾ تقريباً ساڳيو سينٽيڪس استعمال ڪبو آهي.
هاڻي اچو ته هنن آپريٽرز جو استعمال ڪرڻ سکون.

پلس آپريٽر Plus operator :

هي آپريٽر ٻن عددن يا ويري ايبلز جي جوڙ لاءِ استعمال ٿيندو آهي.

پروگرام نمبر 11

```
#include<constream.h>
int main()
{
clrscr();
int a,b,c;
cout<<"Enter 1st value"<<endl;
cin>>a;
cout<<"Enter 2nd value"<<endl;
cin>>b;
cout<<"Enter 3rd value"<<endl;
cin>>c;
cout<<"Addition is "<<a+b+c;
getch();
return 0;
}
```

هن پروگرام ۾ اوهان کي ويري ايبل a,b,c جو جوڙ ڏيکاريو ويندو.

مائينس آپريٽ minus operaters ؛

هي آپريٽر بن عددن جي ڪٽ لاءِ استعمال ٿئي ٿو.

پروگرام نمبر 12

```
#include<constream.h>
int main()
{
clrscr();
int a,b;
cout<<"Enter 1st value"<<endl;
cin>>a;
cout<<"Enter 2nd value"<<endl;
cin>>b;
cout<<"subtraction is "<<a-b;
getch();
return 0;
}
```

هن پروگرام ۾ ويري ايبل A مان B جي ويليو ڪٽ ڪئي ويندي.

ملٽيپلڪيشن آپريٽر Multiplction operator؛

اسان وٽ ٽيون ارٿ ميٽڪ آپريٽر آهي ملٽيپڪيشن جنهن جو ڪم بن انگن کي ضرب ڪرڻ آهي.

پروگرام نمبر 13

```
#include<constream.h>
int main()
{
clrscr();
int a,b;
cout<<"Enter 1st value"<<endl;
cin>>a;
cout<<"Enter 2nd value"<<endl;
cin>>b;
cout<<"subtraction is "<<a*b;
getch();
return 0;
}
```

هن پروگرام ۾ A ۽ B جي ويليوز کي پاڻ ۾ ضرب ڪيو ويندو.

ڊويزن آپريٽر Division operator؛

ڊويزن ميٿ وانگر سي پلس پلس ۾ پڻ ساڳيو ڪم ڪري ٿو، هي پهريون رقم کي مخصوص حصن ۾ ورهائڻ لاءِ ڪم اچي ٿو.

پروگرام نمبر 14

```
#include<constream.h>
int main()
{
clrscr();
int a,b;
cout<<"Enter 1st value"<<endl;
cin>>a;
cout<<"Enter 2nd value"<<endl;
cin>>b;
cout<<"subtraction is "<<a/b;
getch();
return 0;
}
```

ريمائينڈر آپريٽرRemainder operator:

ريمائينڈر آپريٽر انهي لاءِ ڪم ايندو آهي ته اسان جڏهن بن عددن کي هڪ ٻئي مان ونڊ ڪندا آهيون، جيڪا به پاڻي بچندي آهي، انهي کي ڊيڪارڊ لاءِ اسان هي آپريٽر استعمال ڪندا آهيون.

پروگرام نمبر 15

```
#include<constream.h>
int main()
{
clrscr();
int a,b;
cout<<"Enter 1st value"<<endl;
cin>>a;
cout<<"Enter 2nd value"<<endl;
cin>>b;
cout<<"subtraction is "<<a/b;
cout<<”remainder is “<<a%b;
getch();
return 0;
}
```

اوهان کي هاڻي ارٿ ميٽڪ آپريٽر جو استعمال باآساني اچي ويو هوندو، اوهان سوچيندا هوندا ته هنن پنجن پروگرامن بابت اوهان کي ڇو نه سمجهايو ويو. دراصل ڳالهه هي آهي ته اوهان کي هنن پروگرامن جي اسٽيڪچر جي باري ۾ اڳئي سمجهائي آيو آهيان، هنن ۾ صرف آپريٽر جي تبديلي ڪئي وئي آهي. جيڪڏهن اوهان کي اجان به هنن پروگرامن جي سمجهڻ ۾ مشڪل ٿي رهي آهي ته هن انسٽريڪشنز کي وري پڙهو.

ارٿ ميٽڪ اسائيمنٽ آپريٽرArithmetic assignment operators:

سي پلس پلس ۾ اوهان ڪيترن ئي مختلف طريقن سان ڪوڊنگ لکي پنهنجي پروگرام جو سائيز ۽ ڪوڊنگ کي گھٽ ڪري سگھو ٿا، هتي اسان اهڙي ئي آپريٽر جو ذڪر ڪنداسين جيڪو ارٿ ميٽڪ آپريٽر ۽ اسائيمنٽ آپريٽر کي گڏي هڪ نئون آپريٽر ارٿ ميٽڪ اسائيمنٽ آپريٽر ناهيو ويو آهي.

اچو ٽ هڪ مثال ونون.

```
a=a+b;
```

اسان مٿين لاجڪ کي ارٿ ميٽڪ اسائيمنٽ ذريعي هن طرح سان لکنداسين.

```
a+=b
```

ڪجھ ٻيا ٻ ارٿ ميٽڪ اسائيمنٽ آپريٽر آهن، جن کي اوهان مٿيئن طريقي سان پروگرام ۾ استعمال ڪري سگھو ٿا.

-=	*=	/=	%=

اچو ٽ هاڻي ارٿ ميٽڪ اسائيمنٽ اپريٽر جو پروگرام سکون.

پروگرام نمبر 16

```
#include<constream.h>
int main()
{
clrscr();
int ans=27;
ans+=10;
cout<<ans<<" , ";
ans-=7;
cout<<ans<<" , ";
ans*=2;
cout<<ans<<". ";
ans/=3;
cout<<ans<<" , ";
ans%=3;
cout<<ans<<endl;
getch();
return 0;
}
```

هن پروگرام جو آئوٽ پٽ هن ريت نڪرندو.



اچو ٽ ڏسون ٽ هي آئوٽ پٽ ڪيئن آيو آهي.

اسان سڀ کان پهريان هڪ ويري ايبل ڊڪليئر ڪيو آهي، جنهن جو نالو اسان ans ڏنو آهي، ۽ اسائيمنٽ آپريٽر ذريعي ان جي ويو 27 مقرر ڪيوسين.

ٻئي نمبر ۾ اوهان کي جيڪو ans+=10; نظر اچي رهيو آهي ان کي سي پلس پلس جي ٻولي ۾ لاجڪ Logic چيو وڃي ٿو.

اسان ويري ايبل ans جي ويليو 27 ڏئي آيا آهيون، ارٿ ميٽڪ اسائيمنٽ آپريٽر انهن 27 ۾ 10 جوڙ ڪري ans جي ويليو 37 ڪندو.

پوءِ cout جي استيٽمنٽ ڏنل آهي، cout جو ڪم آهي ته ans جي ويليو اسڪرين تي پرنٽ ڪري، آخر ۾ انسرشن آپريٽر کان پوءِ ڪاما (,) ڏنل آهي، جيڪو پڻ ans جي ويليو کان پوءِ پرنٽ ٿيندو.

وري به لاجڪ استعمال ٿيل آهي، ans-=7 ڏنل آهي، هي ans جي ويليو يعني 37 مان 7 ڪٽ ڪندو، ۽ وري سي آئوٽ ans جي ويليو 30 پرنٽ ڪندو.

ان کان پوءِ وري به لاجڪ استعمال ٿيل آهي، ans *=2 هاڻي ans جي ويليو يعني 30 کي 2 سان ضرب ڪندو، ۽ وري سي آئوٽ ans جي ويليو 60 پرنٽ ڪندو.

هاڻي وري به لاجڪ استعمال ٿيل آهي، ans/=3 هاڻي ans جي ويليو يعني 60 کي 3 سان وند ڪندو، ۽ سي آئوٽ ans جي ويليو 20 اسڪرين تي پرنٽ ڪندو.

هاڻي ڏسو آخري لاجڪ ans%=3 هاڻي ans جي ويليو يعني 20 کي 3 سان وند ڪندو ۽ بچيل پاڇي يعني 2 کي سي آئوٽ جي مدد سان اسڪرين تي پرنٽ ڪرائيندو.

تمام آساني سان پروگرام نمبر13 بابت ٻڌايل آهي، مون کي يقين آهي ته اوهان کي آساني سان پروگرام سمجهڻ ۾ اچي ويو هوندو.

باب ڇهون
رليشن آپريٽرس

Relation operator

رليشن آپريٽر بن ويليوز کي پاڻ ۾ ڪمپيئر ڪندا آهن، اهي ويليو ڪهڙن ئي ڊيٽا ٽائيپز ۾ ڏنل هجن مثال: انٽيجر، ڪريڪٽر، فلوٽ وغيره.
سي پلس پلس ۾ ڇهه رليشن آپريٽرز آهن، جن جي لسٽ هيٺ ڏجي ٿي.

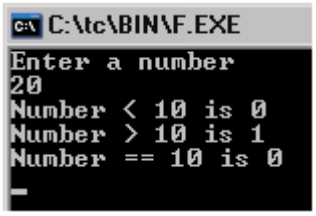
رليشن آپريٽرز	رليشن آپريٽرز جا نالا
>	گريٽر دين (greater than)
<	ليس دين (less than)
==	ايقيول ٽو (equal to)
!=	ناٽ ايقيول ٽو (not equal to)
>=	گريٽر دين آر ايقيول ٽو (greater than or equal to)
<=	ليس دين آر ايقيول ٽو (less than or equal to)

جڏهن اسان رليشن آپريٽرز جي ذريعي بن ويليو کي پاڻ ۾ ڪمپيئر ڪندا آهيون، جڏهن ڪنڊيشن صحيح هجي مثال: 1> 10، چا 10 وڌيڪ آهن 1 کان، اسان آساني سان چئي سگهون ٿا ته بلڪل 10 وڌيڪ آهن 1 کان، اها ڪنڊيشن صحيح آهي، انهي لاءِ سي پلس پلس صحيح جي جاءِ تي 1 پرنٽ ڪندو.
پر جي ڪنڊيشن غلط هجي، مثال: 11> 10 چا 10 وڌيڪ آهن 11 کان، اسان آساني سان چئي سگهون ٿا ته 10 وڌيڪ ناهن 11 کان، اها ڪنڊيشن غلط آهي، انهي لاءِ سي پلس پلس غلط جي جاءِ تي 0 پرنٽ ڪندو. اچو ته رليشن آپريٽرز جو هڪ سادو مثال ونون.

پروگرام نمبر 17

```
#include<constream.h>
int main()
{
clrscr();
int numb;
cout<<"Enter a number"<<endl;
cin>>numb;
cout<<"Number < 10 is "<<(numb<10)<<endl;
cout<<"Number > 10 is "<<(numb>10)<<endl;
cout<<"Number == 10 is "<<(numb==10)<<endl;
getch();
return 0;
}
```

هن پروگرام جو آئوٽ پُٽ هن ريت نڪرندو.



اچو ٽہ ڏسون هن پروگرام جي اسٽيڪچر چا آهي.

اسان سپ کان هڪ انٽيجر ويري ايبل ڊڪليئر ڪيو آهي، جنهن جو نالو اسان numb رکيو آهي.

پوءِ اسان هڪ اسٽرينگ ڪانسٽنٽ Enter a number اسڪرين تي پرنٽ ڪرايو آهي. پوءِ اسان cin جي ذريعي numb جي ويليو کي بورڊ مان ورتي آهي، جيڪو اسان 20 ڏنو آهي.

ان کان پوءِ اسان cout جي ذريعي Number < 10 پرنٽ ڪرايو آهي. ۽ پوءِ اسان انسرشن آپريٽر ڏئي ويليو کي ڪمپيئر ڪرايو آهي. (numb < 10) اسان کي خبر آهي ٽہ اسان numb جي ويليو 20 ڏني آهي، هاڻي ڪنڈيش غلط آهي، چو ٽہ 20 وڏيڪ ناهن 10 کان انهي لاءِ استيٽمنٽ جي سامهون 0 پرنٽ ٿيل آهي.

انهي کان پوءِ هي استيٽمنٽ ۾ پهريان اسٽرينگ ڪانسٽنٽ Number > 10 پرنٽ ڪرايل آهي. پوءِ ويليوز کي ڪمپيئر ڪيو ويو آهي، (num > 10) اسان کي آهي ٽہ اسان numb جي ويليو 20 مقرر ڪئي آهي، هاڻي ڪنڈيشن صحيح آهي، چو ٽہ 20 وڏيڪ آهن 10 کان ناهي لاءِ استيٽمنٽ جي سامهون 1 پرنٽ ٿيندو.

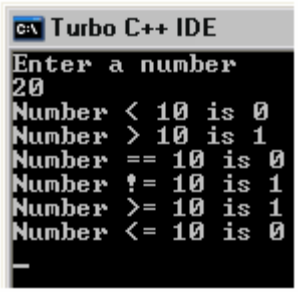
آخري استيٽمنٽ ۾ ساڳي طرح پهريان هڪ استيٽمنٽ Number == 10 پرنٽ ڪرايل آهي، پوءِ ويليوز کي ڪمپيئر ڪرايل آهي، (numb == 10) اسان کي خبر آهي ٽہ numb جي ويليو اسان 20 مقرر ڪئي آهي، هاڻي ڪنڈيشن غلط آهي، چو ٽہ 20 برابر ناهن 10 جي، انهي لاءِ استيٽمنٽ جي سامهون 0 پرنٽ ٿيندو.

هاڻي اچو ٽہ هڪ اهڙي پروگرام جو مثال ونون جنهن ۾ چھ ئي رليشن آپريٹر استعمال ڪيل آهن.

پروگرام نمبر 18

```
#include<constream.h>
int main()
{
clrscr();
int numb;
cout<<"Enter a number"<<endl;
cin>>numb;
cout<<"Number < 10 is "<<(numb<10)<<endl;
cout<<"Number > 10 is "<<(numb>10)<<endl;
cout<<"Number == 10 is "<<(numb==10)<<endl;
cout<<"Number != 10 is "<<(numb!=10)<<endl;
cout<<"Number >= 10 is "<<(numb>=10)<<endl;
cout<<"Number <= 10 is "<<(numb<=10)<<endl;

getch();
return 0;
}
```

پروگرام نمبر 18 جو آئوٽ پٽ ڪجھ هن ريت ايندو.



هن پروگرام جا پهريان ٽي اسٽيٽمنٽ اوهان کي پروگرام نمبر 17 ۾ ٻڌايل آهن.

چوٿين نمبر اسٽيٽمنٽ ۾ چيل آهي ته (numb =! 10) جيڪا ڪنڊيشن صحيح آهي، چو ته اسان numb جي ويليو 20 پرنٽ ڪرائي آهي، اسان کي خبر آهي ته 20 برابر ناهن 10 جي، انهي لاءِ اسٽيٽمنٽ جي سامهون 1 پرنٽ ٿيندو.

پنجين نمبر اسٽيٽمنٽ ۾ چيل آهي ته (numb > =10) جيڪا ڪنڊيشن صحيح آهي، چو ته اسان کي خبر آهي ته numb جي ويليو 20 مقرر ڪيل آهي، اسان ڄاڻو ٿا ته 10 برابر ناهن 20 جي، پر 20 وڏيڪ آهن 10 کان انهي لاءِ اسٽيٽمنٽ جي سامهون 1 پرنٽ ٿيل آهي.

آخري اسٽيٽمنٽ ۾ چيل آهي ته (numb < = 10) جيڪا ڪنڊيشن غلط آهي، چو ته اسان کي خبر آهي ته numb جي ويليو 20 مقرر ڪيل آهي، اسان ڄاڻو ٿا ته 20 نه برابر آهن 10 جي ۽ نه ئي گهٽ آهن 10 کان، انهي لاءِ اسٽيٽمنٽ جي سامهون 0 لکيل آهي.

هاڻي اوهان رليشن آپريٽر جي استعمال ۽ ڪم کان واقف ٿي ويا هوندا.

باب ستون
لوپس
Loops

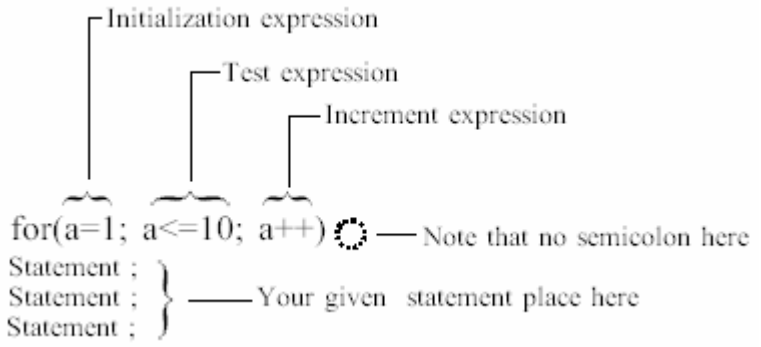
اوهان انگلش لفظ Loop کان اڳئي واقف هوندا، جنهن جي معنيٰ آهي مسلسل هلڻ واري شئي.
ساڳئي طرح سي پلس پلس ۾ به لوپ جو ساڳيو ئي ڪانسيپٽ Concept آهي. اسان هڪ استيٽمنٽ جي ذريعي گهڻيون ئي استيٽمنٽس هڪ جڳهه تي پرنٽ ڪرائي سگهون ٿا.
اسان پروگرام ۾ لوپ جو استعمال پروگرام جي ڪوڊنگ گهٽائڻ ۽ سائيز گهٽائڻ لاءِ استعمال ڪندا آهيون، لوپ جي ذريعي نهيل پروگرام سمجهڻ ۾ پڻ آسان هوندو آهي.
سي پلس پلس ۾ لوپ جا ٽي قسم آهن.

1. فار لوپ (for loop)
2. وائيل لوپ (while loop)
3. ڊو وائيل لوپ (do while loop)

هاڻي اچو ته لوپ جي قسمن بابت ڪجهه سکون ۽ انهن جي باري ۾ ڪجهه مثال ڏسون.

1. فار لوپFor loop:

اسان فار لوپ ان وقت استعمال ڪندا آهيون جڏهن اسان کي خبر هجي ته اسان کي گهٽا چڪر استيٽمنٽ پرنٽ ڪرائڻي آهي، مثال: اسان کي Sindh لفظ 10 چڪر پرنٽ ڪرائڻو آهي ته اسان فار لوپ استعمال ڪندا سين.
اچو ته فار لوپ جي سانٽيڪس لکڻ سکون.



فگر نمبر 3

اوهان کي مٿي فار لوپ جي استيڪچر نظر اچي رهي آهي، اچو ته هاڻي فار لوپ کي سمجهڻ جي ڪوشش ڪريون.

Initialization expression اسان هن ۾ ويري ايبل جي پهريان ويليو ڏيندا آهيون، جيئن فگر نمبر3 ۾ ويري ايبل a جي ويليو 1 ڏنل آهي. هي صرف هڪ چڪر هلندو آهي پوءِ لوپ test expression ۽ increment expression ڏانهن رجوع ڪندو آهي.

Test expression اسان هن ۾ رليشن آپريٽر جي ذريعي ويليوز کي ڪمپيئر ڪرائيندا آهيون، جڏهن ويليو غلط ٿي ويندو ته لوپ ختم ٿي ويندو.

Increment expression دراصل هن ۾ ويري ايبل جي ويليو 1 وڌندي آهي، جيئن اسان a جي ويليو 1 ڏني آهي، جڏهن لوپ استيٽمنٽ پرنٽ ڪندو ته وري Increment expression ڏانهن ويندو، Increment expression ويري ايبل a جي ويليو 2 ڪري ڇڏيندو. اسان لوپ ۾ Increment expression جي جڳهه تي decrement expression به استعمال ڪندا آهيون، هي ويري ايبل جي ويليو مان 1 ڪٽ ڪندو آهي.

ياد رکو ته فار لوپ جي اڳيان سيمي ڪالن استعمال نه ٿيندو ۽ جي اوهان سيمي ڪالن استعمال ڪندا ته پوءِ اوهان جو پروگرام ته هلي سگهي ٿو پر آئوٽ پٽ صحيح نه ايندو.

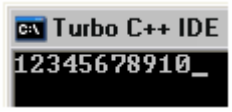
هميشه جيڪا به استيٽمنٽ اوهان کي پرنٽ ڪرائڻي آهي، ته اها فار لوپ جي اسٽڪچر جي هينيان لکڻ گهرجي.

اچو ته فار لوپ ۾ هڪ سادي پروگرام جو مثال ونون.

پروگرام نمبر 19

```
#include<constream.h>
int main()
{
clrscr();
int a;
for(a=1; a<=10; a++)
cout<<a;
getch();
return 0;
}
```

هن پروگرام جو آئوٽ پٽ هن ريت نڪرندو.



اچو ته ڏسون هن پروگرام جو آئوٽ پٽ ڪيئن نڪتو.

سڀ کان پهريان هڪ انٽيجر ويري ايبل a ڊڪليئر ڪيل آهي. پوءِ فار لوپ جي اندر انٽيجر ويري ايبل a جي ويليو 1 مقرر ڪيل آهي.

هاڻي ڪمپائيلر ٽيسٽ ايڪسپريشن ڏانهن ويندو.

```
for(a=1; a<=10; a++)
```

پوءِ ڪمپائيلر ڏسندو ته واقعي 1 ننڍو آهي 10 کان، جي ها 1 ننڍو آهي 10 کان، انهي لاءِ ڪنڊيشن صحيح آهي، ڪمپائيلر هيٺ ايندو ۽ استيٽمنٽ پرنٽ ڪندو.

```
for(a=1; a<=10; a++)
cout<<a;
```

هاڻي cout ويري ايبل a جي ويليو پرنٽ ڪندو يعني 1.

پوءِ وري ڪمپائيلر انڪريمنٽ ايڪسپريشن ڏانهن ويندو.

```
for(a=1; a<=10; a++)
cout<<a;
```

انڪريمنٽ ايڪسپريشن ويري ايبل a جي ويليو1 مان وڌائي 2 ڪندو.

۽ پوءِ وري ڪمپائيلر ٽيسٽ ايڪسپريشن ڏانهن ويندو.

```
for(a=1; a<=10; a++)
cout<<a;
```

چا واقعي 2 ننڍو آهي 10 کان، ڪنڊيشن صحيح آهي، انهي لاءِ ڪمپائيلر وري استيٽمنٽ پرنٽ ڪرائيندو.

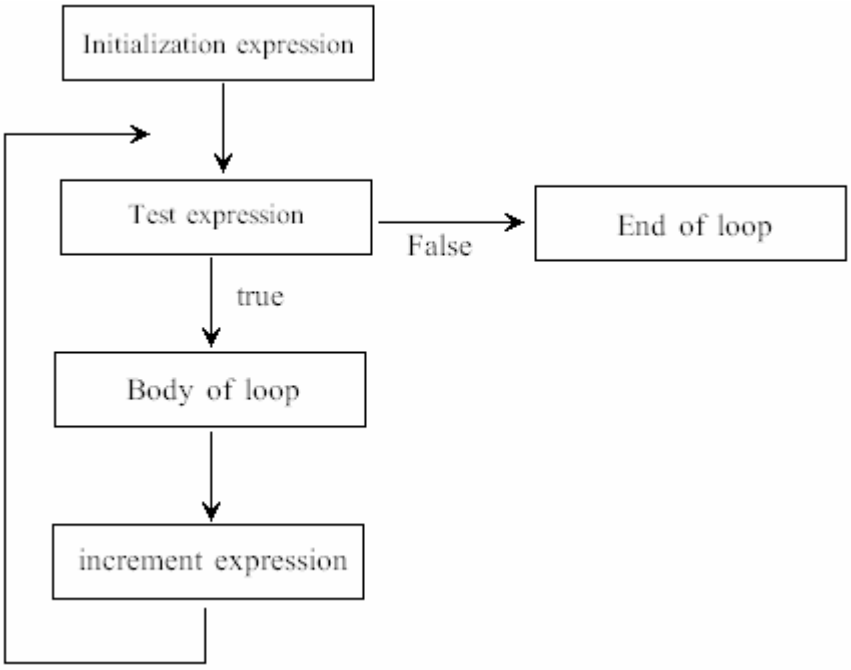
```
for(a=1; a<=10; a++)
cout<<a;
```

هاڻي a جي ويليو 2 آهي، cout هاڻي 2 پرنٽ ڪندو.

اهو سلسلو هلندو رهندو، وري ڪمپائيلر ويندو انڪريمنٽ ايڪسپريشن ڏانهن، انڪريمنٽ ايڪسپريشن a جي ويليو 3 ڪندو، وري ڪمپائيلر ٽيسٽ ايڪسپريشن ڏي ويندو، جيڪڏهن ڪنڊيشن صحيح آهي ته وري استيٽمنٽ پرنٽ ڪرائيندو.

لوپ هلندي آخر ۾ a جي ويليو وڌندي 10 ٿيندي ۽ پوءِ ڪمپائيلر ٽيسٽ ايڪسپريشن ڏانهن ويندو، 10 گهٽ ناهن 10 جي، پر 10 برابر آهن 10 جي، انهي لاءِ هي ڪنڊيشن صحيح آهي ڪمپائيلر استيٽمنٽ پرنٽ ڪرائيندو.

وري ڪمپائيلر انڪريمنٽ اڪسپريشن ڏانهن ويندو، انڪريمنٽ آپريٽر a جي ويليو 11 ڪندو، ڪمپائيلر وري ٽيسٽ ايڪسپريشن ڏانهن ويندو، ڏسندو ته 11 برابر آهن 10 جي يا گهٽ آهن 10 کان، ٻئي ڪنڊيشن غلط آهن انهي لاءِ لوپ ختم ٿي ويندو ۽ پروگرام به ختم ٿي ويندو.



فگر نمبر4

اوهان کي فگر نمبر 4 ۾ فار لوپ جي سڄي اسٽيڪچر ٻڌايل آهي، سي کان پهريان انيشلائيزيشن پوءِ ٽيسٽ ايڪسپريشن، ڪنڊيشن غلط هجي ته لوپ ختم ۽ جي ڪنڊيشن صحيح هجي ته اسٽيٽمنٽ پرنٽنگ، ۽ پوءِ وري انڪريمنٽ ايڪسپريشن، ۽ وري ساڳيو چڪر شروع.

ڊڪريمنٽ ايڪسپريشنDecrement expression:

اسان مٿي انڪريمنٽ ايڪسپريشن متعلق سکي آيا آهيون، انڪريمنٽ ايڪسپريشن ويري ايبل جي ويليو ۾ 1 جو اضافو ڪندو آهي.

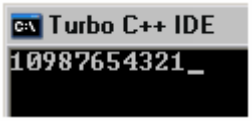
ساڳئي طرح ڊڪريمنٽ ايڪسپريشن ويري ايبل جي ويليو مان 1 ڪٽ ڪندو آهي.

اچو ته ڊڪريمنٽ ايڪسپريشن جو هڪ پروگرام ڏسون.

پروگرام نمبر 20

```
#include<constream.h>
int main()
{
clrscr();
int a;
for(a=10; a>=1; a--)
cout<<a;
getch();
return 0;
}
```

پروگرام نمبر 17 جو آئوٽ پٽ ڪجهه هن ريت ايندو.



اچو ته ڏسون ته هن پروگرام جو آئوٽ پٽ ڪيئن نڪتو.
اوهان کي فار لوپ جي سڃي اسٽيڪچر ۽ ڪم پروگرام نمبر 19 ۾ ٻڌايل آهي. سڀ کان پهريان ڪمپائيلر انيشلائيزيشن ڏي ويندو، جتي ويري ايبل a جي ويليو 10 مقرر ڪيل آهي، پوءِ وري ڪمپائيلر ٽيسٽ ايڪسپريشن ڏانهن ويندو، a>=1 چا 10 وڏا آهن يا برابر آهن 1 جي، 10 هڪ جي برابر ته ناهي پر 1 کان وڏو آهي، انهي لاءِ ڪنڊيشن صحيح آهي، ڪمپائيلر سي آئوٽ ڏانهن ويندو ۽ ويري ايبل a جي ويليو 10 پرنٽ ڪرائيندو. پوءِ وري ڪمپائيلر ڊڪريمنٽ ايڪسپريشن ڏانهن ويندو، ۽ a جي ويليو يعني 10 مان 1 ڪٽ ڪندو ۽ a جي ويليو 9 ڪندو. وري ٽيسٽ ايڪسپريشن ڏانهن ويندو، ڪنڊيشن صحيح هوندي ته هيٺ سي آئوٽ ڏانهن ايندو، ۽ a جي ويليو يعني 9 پرنٽ ڪرائيندو.
آخر ۾ ڊيڪريمنٽ ايڪسپريشن جي مدد سان a جي ويليو گهٽ ٿيندي 1 ۾ اچي بيهندي، پوءِ ڪمپائيلر وري ٽيسٽ ايڪسپريشن ڏانهن ويندو، چا 1 برابر آهن 1 جي، يا وري 1 وڏو آهي 1 کان. 1 وڏو ناهي 1 کان، پر 1 برابر آهي 1 جي، انهي ڪنڊيشن صحيح آهي، ڪمپائيلر هيٺ ايندو، ۽ a جي ويليو يعني 1 پرنٽ ڪندو، وري ڪمپائيلر متي ڊڪريمنٽ ايڪسپريشن ڏانهن ويندو، ۽ a جي ويليو کي 1 مان گهٽائي 0 ڪندو، پوءِ ڪمپائيلر ٽيسٽ ايڪسپريشن ڏانهن ويندو ۽ ڏسندو ته 0 برابر آهي 1 جي، يا 0 وڏو آهي 1 کان. هتي ٻئي ڪنڊيشن صحيح ناهن، چو ته 0 نه ته 1 کان وڏو آهي ۽ نه ئي برابر، انهي لاءِ لوپ ختم ٿي ويندو.

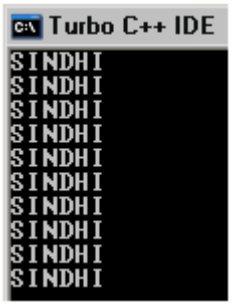
فار لوپ استيٽمنٽfor loop statement:

اوهان هڪ مثال ونو اوهان کي 10 چڪر لفظ SINDHI پرنٽ ڪرائڻو آهي، اوهان جيڪڏهن لوپ کان بغير اهو پرنٽ ڪرائيندا ته اوهان کي هن طرح پروگرام لکڻو پوندو.

پروگرام نمبر 21

```
#include<constream.h>
int main()
{
clrscr();
cout<<"SINDHI"<<endl;
cout<<"SINDHI"<<endl;
cout<<"SINDHI"<<endl;
cout<<"SINDHI"<<endl;
cout<<"SINDHI"<<endl;
cout<<"SINDHI"<<endl;
cout<<"SINDHI"<<endl;
cout<<"SINDHI"<<endl;
cout<<"SINDHI"<<endl;
cout<<"SINDHI"<<endl;
getch();
return 0;
}
```

هن پروگرام جو آئوٽ پٽ ڪجهه هن ريت نڪرندو.

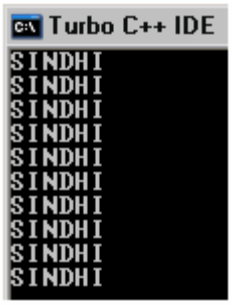


پر اسان اهو ساڳيو پروگرام فار لوپ ۾ لکنداسين تہ اهو ننڍو ۽ آسان بيهندو ۽ آئوٽ پٽ ۾ ساڳئي نڪرندي.

پروگرام نمبر 22

```
#include<constream.h>
int main()
{
clrscr();
int a;
for(a=1; a<=10; a++)
cout<<"SINDHI"<<endl;
getch();
return 0;
}
```

هن پروگرام جو ۾ آئوٽ پٽ 21 نمبر پروگرام جهڙو ئي نڪرندو.



هن پروگرام کي سمجهڻ لاءِ پروگرام نمبر 19 کي ڏسو، هن پروگرام ۽ پروگرام نمبر 19 ۾ هڪ فرق آهي، پروگرام نمبر 19 ۾ اسان سي آئوٽ ذريعي ويري ايبل جي ويليو مثال: 12345678910 پرنٽ ڪرائيندا هئاسين، هاڻي ان جڳهه تي اسان اسٽرنگ ڪانسٽنٽ sindhi کي پرنٽ ڪرايو آهي.

ادر استيٽمنٽ other statement:

اسان سي پلس پلس ۾ مختلف قسم جا استيٽمنٽ پرنٽ ڪرائي سگهون ٿا. اڃو هڪ استيٽمنٽ جو مثال ونون جنهن ۾ ارٽ ميٽڪ جو استعمال ٿيل آهي.

پروگرام نمبر 23

```
#include<constream.h>
int main()
{
clrscr();
int a;
for(a=1; a<=10; a++)
cout<<a+a<<" ";
getch();
return 0;
}
```

هن پروگرام جو آئوٽ پٽ ڪجهه هن ريت نڪرندو.



پروگرام نمبر 23 جو استيڪچر مٿي لوپ بابت ڏنل مثالن جهڙو ئي آهي. هن پروگرام ۾ صرف هڪ فرق آهي ته اڳئي اسين سي آئوٽ ذريعي استيٽمنٽ پرنٽ ڪرائيندا هئاسين، پر پروگرام نمبر 23 ۾ اسان سي آئوٽ ۾ ويري ايبل a کي a سان جوڙ ڪرايو آهي، پهريان ويري ايبل جي ويليو 1 مقرر ڪيل آهي، سي آئوٽ 1+1 جو جوڙ 2 پرنٽ ڪرائيندو. وري ڪمپائيلر مٿي انڪرمنٽ آپريٽر ڏانهن ويندو، انڪريمنٽ آپريٽر وري ايبل a جي ويليو ۾ ڪندو، سي آئوٽ 2+2 جو جوڙ 4 پرنٽ ڪرائيندو. ان کان پوءِ ڊبل ڪوٽيشن جي اندر هڪ اسپيس انهي لاءِ ڇڏيل آهي ته جيئن انگن جي وچ ۾ اسپيس پرنٽ ٿئي.

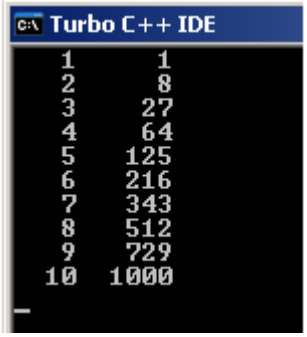
ملٽي پُل اسٽيٽمنٽ ان لوپ بادي Multipul statement in loop body:

يقينن اوهان فار لوپ ۾ هڪ کان وڌيڪ اسٽيٽمنٽ پرنٽ ڪرائڻ جا خواهشمند هوندا. جي ها! اوهان فار لوپ ۾ هڪ کان وڌيڪ اسٽيمنٽ پرنٽ ڪرائي سگهو ٿا. اوهان کي لوپ بادي ۾ ڏنل اسٽيٽمنٽ کي بريسز ڏيڻو پوندو. اچو ته هڪ مثال وٺون جنهن ۾ هڪ پاسي انگ ڏنل آهن ۽ ان جي سامهون انهن جا ڪيوب ڏيکاريل آهن.

پروگرام نمبر 24

```
#include<constream.h>
#include<iomanip.h>
void main()
{
clrscr();
for(int numb=1; numb<=10; numb++)
{
cout<<setw(4)<<numb;
int cube=numb*numb*numb;
cout<<setw(6)<<cube<<endl;
}
getch();
}
```

هن پروگرام جو آئوٽ پٽ هن ريت نڪرندو.



اچو ته سمجهڻ جي ڪوشش ڪريون ته هن پروگرام جو آئوٽ پٽ ڪيئن نڪتو.
اوهان کي ڪونسٽريم جي هينيان آيو مينپ هيڊر فائيل ڏسڻ ۾ اچي رهيو هوندو. اسين مٿي ذڪر ڪري آيا آهيون ته ايو مينپ سيٽ ڊبليو جي لاءِ استعمال ڪيو ويندو.

وائيوڊ Void:

آيومينپ کان پوءِ اوهان کي هڪ نئون کي ورڊ وائيوڊ نظر اچي رهيو آهي. وائيوڊ کي ورڊ اسان انهي لاءِ استعمال ڪندا آهيون ته هن ۾ ڪو به ريٽن ٽائيپ ناهي، اسان جڏهن int استعمال ڪندا هئاسين ته لازمي اسان کي return 0; استعمال ڪرڻو پوندو هو پر وائيوڊ جي استعمال کان پوءِ اسان کي return 0; جي استعمال مان چوٽڪارو ملي ويندو.

ويري ايبل اِن فار لوپVariable in for loop:

هن پروگرام ۾ ويري ايبل کي ڊڪليئر ڪرڻ جو هڪ نئون طريقو ڏسي رهيا آهيون، اسان اڳي لوپ جي ٻاهران ويري ايبل کي ڪجهه هن ريت ڊڪليئر ڪندا هئاسين int a; پر هن پروگرام ۾ اسان ويري ايبل کي فار لوپ جي اندر ڊڪليئر ڪيو آهي. for(int numb=1; هن استيٽمنٽ ۾ اسان فار لوپ جي اندر ويري ايبل کي ڊڪليئر ڪري ان جي ويليو 1 رکي آهي.

انهي کان پوءِ وري بريسز استعمال ٿيل آهن، بريسز جي اندر ڏنل استيٽمنٽ کي استيمنٽ بادي آف فار لوپ چيو وڃي ٿو، بريسز جي استعمال کان سواءِ اوهان پروگرام ۾ ايرر کي منهن ڏيڻو پوندو.

ان کان پوءِ cout<<setw(4)<<numb; ان جو مطلب آهي ته ويري ايبل numb جي ويليو 4 ڪريڪٽر ڇڏي پوءِ پرنٽ ڪرائي. اوهان مٿي پروگرام جي آئوٽ پُٽ ۾ ڏسي سگهو ٿا ته نمبر 1،2،3 وغيره ٿوري جڳهه يعني 4 ڪريڪٽر ڇڏي پوءِ پرنٽ ٿيل آهن

ان کان پوءِ cube جي نالي سان هڪ ويري ايبل ڊڪليئر ڪيل آهي، ۽ ويري ايبل cube برابر آهي numb*numb*numb جي برابر آهي. مثال نمبر 5 ۾ ڪيوب جي ويليو 125 پرنٽ ڪيل آهي. ڇو ته 5*5*5 هن جي ضرب اپت 125 ٿيندي.
هن ٽائيپ لکيل ڪوڊ کي لاجڪ چئبو آهي.

ان کان پوءِ cout<<setw(6)<<cube; لکيل آهي، هن جو مطلب آهي ته وري ايبل cube جي جيتري به ويليو هجي اها ڇهه ڪريڪٽر ڇڏي پوءِ پرنٽ ڪرائي .

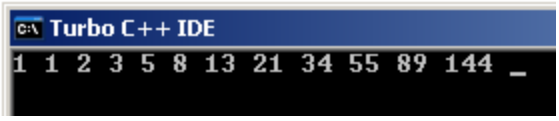
ملٽي پُل انيشلائيزيشنMultipul initialization:

اسان اڳئي هڪ انيشلائيزيشن لکندا هئاسين، مثال for(a=1..... پر هاڻي اسان هڪ وڌيڪ انيشلائيزيشن جو هڪ مثال ونون ٿا.

پروگرام نمبر 25

```
#include<constream.h>
void main()
{
clrscr();
for(int a=1, b=1; a<=100; a=a+b, b=a+b)
cout<<a<<" "<<b<<" ";
getch();
}
```

هن پروگرام جو آئوٽ پٽ ڪجهه هن ريت نڪرندو.



هن سيريز کي فائيبونڪ سيريز چيو وڃي ٿو. هن جي آئوٽ ۾ ڏسندا ته پهريان بن عددن جو جوڙ پرنٽ ڪرايل آهن مثال 1+1=2 پوءِ وري آخري 1 ۽ 2 جو جوڙ 3 پرنٽ ڪرايل آهي. پوءِ 2۽ 3 جو جوڙ 5 پرنٽ ڪرايل آهي.

اچو هاڻي هن پروگرام جي آئوٽ پُٽ تي ويچار ڪريون.

هن پروگرام جي انيشلائيزيشن ۾ ٻه وري ايبل a ۽ b ڊڪليئر ڪيل آهن، ٻنهي جي ويليو 1 مقرر ڪيل آهي.

پوءِ ٽيسٽ ايڪسپريشن ڏنل آهي.

پوءِ انڪريمنٽ آپريٽر جي جڳهه تي a=a+b, b=a+b ڏنل آهي، آئون سمجهان ٿو ته مٿي پروگرام ۾ سيئي ڪوڊنگ اوهان کي سمجهايل آهن سواءِ انڪريمنٽ آپريٽر جي جڳهه تي استعمال ٿيل a=a+b, b=a+b جي باري ۾ اوهان کي مٿي نه ٻڌايو ويو آهي.

جڏهن ڪمپائيلر هڪ چڪر سي آئوٽ طرف ويندو، پوءِ وري اينڪريمنٽ آپريٽر وٽ ايندو. پهريان ڪمپائيلر b=a+b طرف ويندو، a ۽ b جي ويليو 1 آهن ٻنهي جي جوڙ يعني 2 کي b ۾ سيو ڪندو. هاڻي ويري ايبل b جي ويليو 2 ٿي ويندي، پوءِ ڪمپائيلر وري a=a+b ڏانهن ويندو. اسان کي خبر آهي ته a جي ويليو 1 مقرر ڪيل آهي ۽ هاڻي b جي ويليو 2 آهن انهن ٻنهي جو جوڙ يعني 3 کي ويري ايبل a ۾ سيو ڪندو ۽ هيٺ پرنٽ ڪندو، جيسيتائين ڪنڊيشن صحيح هوندي لوپ هلندو رهندو.

سي آئوٽ وري ايبل a ۽ b جي ويليو پرنٽ ڪرائيندو ويندو.

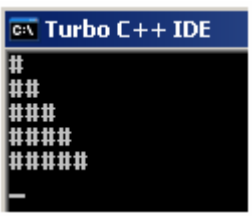
نيسٽيڊ فار لوپ Nestes for loop:

نيسٽ معنيٰ جار ۽ نيسٽيڊ لوپ معنيٰ، جار لوپز جو. اسان نيسٽيڊ فار لوپ ۾ هڪ کان وڌيڪ فار لوپ استعمال ڪنداسين. اچو ته نيسٽيڊ فار لوپ جو مثال وٺي ان کي سمجهڻ جي ڪوشش ڪريون.

پروگرام نمبر 26

```
#include<constream.h>
void main()
{
clrscr();
for(int a=1; a<=5; a++)
{
for(int b=1; b<=a; b++)
cout<< "#";
cout<<endl;
}
getch();
}
```

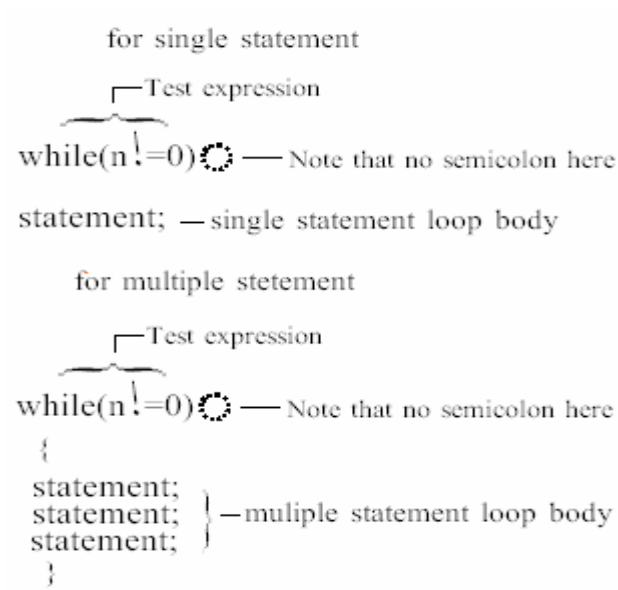
هن پروگرام جو آئوٽ پٽ ڪجهه هن ريت نڪرندو.



هاڻي اچو ته معلوم ڪرڻ جي ڪوشش ڪريون ته هي آئوٽ پٽ ڪيئن نڪتو.
سڀ کان پهريان هڪ ويري ايبل a ڊڪليئر ڪيل آهي، ۽ ان جي ويليو 1 ڏنل آهي. پوءِ ٽيسٽ ايڪسپريشن ڏنل آهي. ڇا 1 ننڍو آهي 5 کان، ها ڪنڊيشن صحيح آهي. ته پوءِ ويري ڪمپائيلر ٻئي لوپ ڏانهن ايندو. ۽ پنج چڪر هيش (#) پرنٽ ڪرائيندو. پوءِ وري پهرين لوپ جي انڪريمنٽ ايڪسپريشن ڏانهن ويندو، ۽ پوري وري پهرين لوپ جي ٽيسٽ ايڪسپريشن ڏانهن ويندي جي ڪنڊيشن صحيح آهي ته هيٺين لوپ ڏانهن ايندو ۽ وري هيش پرنٽ ڪرائيندو.
هي پروگرام ان وقت ختم ٿيندو جڏهن ٻئي لوپ ختم نه ٿي وڃن.

2. وائيل لوپ While loop:

اسان فار لوپ ۾ فڪس نمبرن تي مشتمل اسٽيٽمنٽ پرنٽ ڪرائيندا آهيون، پر جڏهن اسان کي هن ڳالهه جي خبر نه هجي ته اسٽيٽمنٽ ڪيترا ڀيرا پرنٽ ڪرائڻو آهي ته پوءِ اسان وائيل لوپ استعمال ڪنداسون.
اچو ته وائيل لوپ جو سينٽيڪس لکڻ سکون.



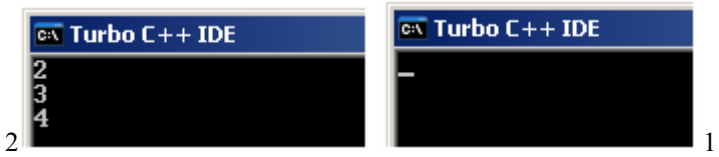
فگر نمبر5

وائيل لوپ ۾ انشلائيزيشن ۽ انڪريمنٽ ايڪسپريشن جو استعمال نه ٿيندو آهي صرف ٽيسٽ ايڪسپريشن جو استعمال ٿيندو آهي.
مٿي فگر نمبر 5 ۾ وائيل لوپ ۾ سنگل ۽ ملٽيپل اسٽيٽمنٽ پرنٽ ڪرائڻ جو سينٽيڪس لکيل آهي.
فگر نمبر 5 ۾ اوهان ڏسندا ته وائيل لوپ جي ٽيسٽ ايڪسپريشن جي سامهون سيمي ڪالن استعمال ٿيل نه آهي. مٿي سنگل ۽ ڊبل اسٽيٽمنٽ پرنٽ ڪرائڻ جو طريقو پڻ ٻڌايل آهي.
اچو ته وائيل لوپ ۾ هڪ پروگرام جو مثال وٺون.

پروگرام نمبر 27

```
#include<constream.h>
void main()
{
clrscr();
int a=99;
while(a!=0)
cin>>a;
cout<<endl;
getch();
}
```

هن پروگرام جو آئوٽ پٽ ڪجهه هن ريت نڪرندو.



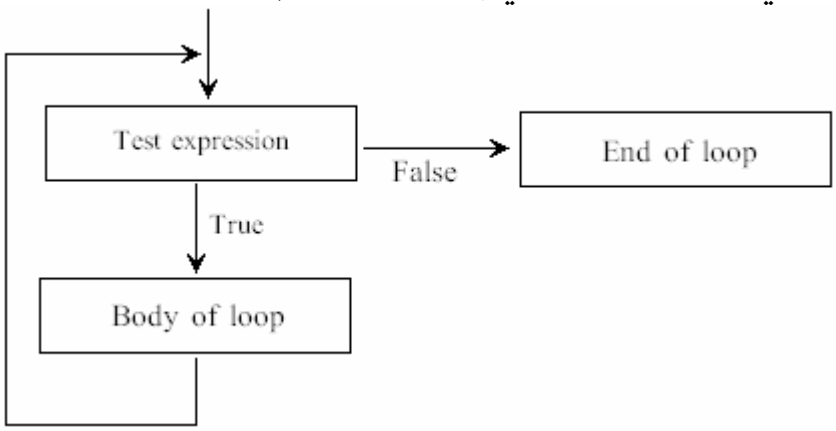
2 1

اچو ته معلوم ڪريون ته هن پروگرام جو آئوٽ پٽ ڪيئن نڪتو.

سڀ کان پهريان اوهان پروگرام نمبر 26 تي غور ڏيو. ان ۾ اوهان کي هڪ انٽيجر a ڊڪليئر ٿيل نظر ايندو. جنهن جي ويليو 99 مقرر ڪيل آهي، اوهان 0 کان سواءِ ڪهڙو به ويليو مقرر ڪري سگهو ٿا.

اوهان کي هن پروگرام ۾ هڪ نئون لائين while(a!=0) لکيل نظر اچي رهيو آهي. جنهن جي معنيٰ آهي ته اسان جو لوپ ايسيتائين هلي جيسيتائين ويري ايبل a جي ويليو 0 نه هجي.

ان کان پوءِ اِن پُٽ فنڪشن سي اِن استعمال ٿيل آهي، جيڪو کي بورڊ کان ويليو وٺندو. اوهان مثال: 2 ڏئي انٽر پريس ڪندا ته ڪرسر هيٺين لائين تي اچي ويندو، پوءِ وري 3،4 پريس ڪري انٽر ڪندا ته وري به ڪرسر هيٺين لائين تي اچي ويندو. پر جڏهن اوهان 0 پريس ڪري انٽر ڪندا ته پروگرام (لوپ) ختم ٿي ويندو. آئوٽ پُٽ 1 ۾ ڪجهه به لکيل ناهي اوهان ڪجهه به لکي پرنٽ ڪريو، لوپ ختم ڪرڻ لاءِ 0 لکو.



فگر نمبر 6

هاڻي اوهان فگر نمبر 6 کي ڏسي وائيل لوپ جي سڄي اسٽيڪچر کان واقف ٿي ويا هوندا.

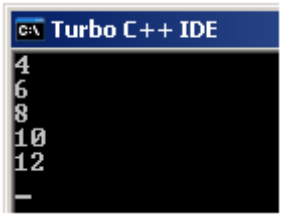
وائيل وِٿ انڪريمنٽ يڪسپريشن :While with increment Exp؛

اسين مٿي لکي آيا آهيون ته وائيل لوپ ۾ انڪريمنٽ ايڪسپريشن استعمال نه ٿيندو آهي، پر اسين هتي دوستن کي ٻڌائيندا هلون ته ڪن ڪنڊيشنن ۾ اسان انڪريمنٽ آپريٽر استعمال ڪري سگهو ٿا، انهي جو هڪ مثال هيٺ ڏجي ٿو.

پروگرام نمبر 28

```
#include<constream.h>
void main()
{
clrscr();
int a=2;
while(a<=10)
{
a+=2;
cout<<a<<endl;
}
getch();
}
```

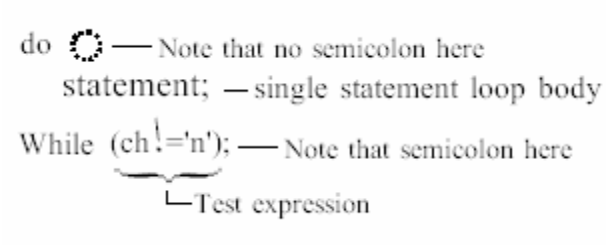
هن پروگرام جو آئوٽ پُٽ ڪجهه هن ريت نڪرندو.



هن پروگرام ۾ a+=2 نڪريمنٽ ايڪسپريشن طور استعامل ٿيل آهي، هن جو مطلب آهي ته ويري ايبل a جي ويليو ۾ 2 جو اضافو ڪيو وڃي. ۽ هي لوپ ايسيتائين هلي جيسيتائين ويري ايبل a جي ويليو 10 جي برابر هجي يا 10 کان گهٽ هجي.

دو- وائيل لوپ Do-while loop؛

اسان وائيل لوپ ۾ پهريان ٽيسٽ ايڪسپريشن رکندا آهيون، جي ڪنڊيشن صحيح هوندي آهي ته پروگرام اڳتي هلندو آهي نه ته پروگرام ختم ٿيو وڃي.
اچو ته دو وائيل لوپ جو سينٽيڪس لکڻ سکون.



فگر نمبر 7

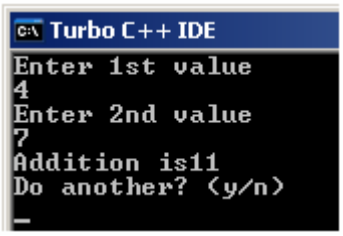
پر ڊو وائيل لوپ ۾ اسان ٽيسٽ ايڪسپريشن رکندا آهيون جي ڪنڊيشن صحيح هوندي ته پروگرام وري هلندو نه ته پروگرام ختم ٿي ويندو.

اچو ته ڊو وائيل لوپ جو هڪ مثال ونون.

پروگرام نمبر 29

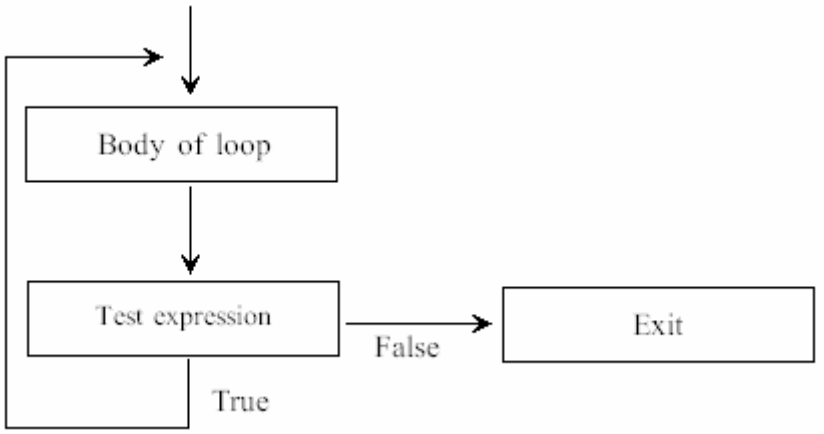
```
#include<constream.h>
void main()
{
clrscr();
int a,b;
char test;
do
{
cout<<"Enter 1st value"<<endl;
cin>>a;
cout<<"Enter 2nd value"<<endl;
cin>>b;
cout<<"Addition is"<<a+b<<endl;
cout<<"Do another? (y/n)"<<endl;
cin>>test;
}
while(test != 'n');
getch();
}
```

هن پروگرام جو آئوٽ پٽ ڪجهه هن ريت نڪرندو.



اچو ته معلوم ڪرڻ جي ڪوشش ڪريون ته هن پروگرام جو آئوٽ پٽ ڪيئن نڪتو.
اوهان کي سڀ کان پهريان ٻه انٽيجر ويري ايبل ۽ هڪ ڪريڪٽر ويري ايبل ڊڪليئر ٿيل نظر اچي رهيا آهن.

اسان ويري ايبل a ۽ b ۾ يوزر کان ويليو ونڊاسين ۽ انهن ۾نهي جو جوڙ سي آئوٽ ذريعي پرنٽ ڪرائينداسين. پوءِ وري هڪ نئون استيٽمنٽ Do another? (y/n) پرنٽ ڪرائينداسين. ان کانپوءِ اسان سي اين جي ذريعي ڪريڪٽر ويري ايبل test ۾ ويليو ونڊاسين. ان کان پوءِ ٽيسٽ ايڪسپريشن ڏنل آهي ته هي لوپ (پروگرام) ان وقت تائين هلي جيستائين Test ويري ايبل ۾ n نه ڏنو وڃي (test برابر هجي n جي) ۽ جڏهن test ويري ايبل ۾ n کان سواءِ ٻيو ڪو الفابيٽ ڏنو وڃي ته لوپ ختم نه ٿئي.



فگر نمبر 8

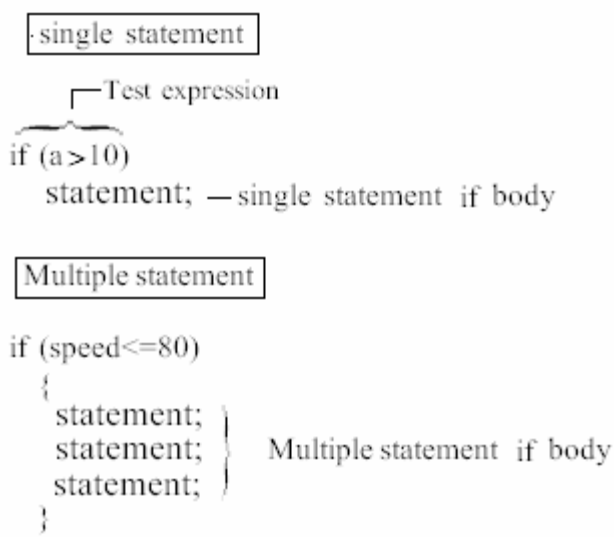
اوهان فگر نمبر 8 ۾ ڏو وائيل لوپ جو ڪم سمجهايل آهي. اوهان فگر 8 کي ڏسي ڏو وائيل لوپ جو ڪم سمجهي سگهو ٿا.

باب اٺون
ڊسيشنس
Decisions

ڊسيشن لفظ جي معنيٰ آهي ته فيصلو ڪرڻ. جيئن اسان ڪنهن ڪلاٿ شاپ ۾ وڃي سوچيندا آهيون ته اسان کي ڪاٽن جو وڳو وٺڻ گهرجي يا کاڌي جو. جيڪڏهن اسان کي هڪ وڳو وٺڻو آهي ته پوءِ لازمي اسان کي فيصلو ڪرڻو پوندو ته هاڻي ڇا وٺجي. مثال اوهان کاڌي ڪپڙو ورتو.
ائين پروگرامنگ ۾ به فيصلي جي ضرورت پوندي آهي. ڪمپائيلر کي ٻڌائڻو پوندو آهي ته هو ڪهڙي ٽائيم تي ڇا پرنٽ ڪري. (جيئن سياري جي موسم ۾ گرم ڪپڙا وٺڻ گهرجن). جيڪڏهن اسان پروگرام ۾ ڊسيشن استعمال نه ڪنداسين ته لازمي آهي ته پروگرام ۾ ڪنڊيشن جي خلاف اسٽيٽمنٽ پرنٽ ٿيندا ۽ يوزر کي تنگ ڪري ماريندا ۽ هو لازمي طور ته پروگرام بند ڪري ڇڏيندو. (جيئن گرمي ۾ گرم ڪپڙا پائي پاڻ کي بيهوش ڪرڻ).
انهي لاءِ عام زندگي وانگر به پروگرامنگ ۾ صحيح فيصلو اڳتي هلي ڪارائتو ثابت ٿيندو.
اچو هاڻي ڏسون ته پروگرامنگ ۾ ڊسيشن ڪيئن وٺجن.

اِف If:

اِف انگلش لفظ آهي جنهن جي معنيٰ جيڪڏهن. اسان عام طور تي به فيصلي ڪرڻ وقت جيڪڏهن لفظ استعمال ڪندا آهيون. مثال: جيڪڏهن بل گيٽس مائيڪرو سافٽ ڇڏي ويندو ته پوءِ اسان مائيڪرو سافٽ سان ڪانٽريڪٽ ٽوڙي ڇڏيندا سين.
اچو ته اِف جو سينٽيڪس لکڻ سکون.



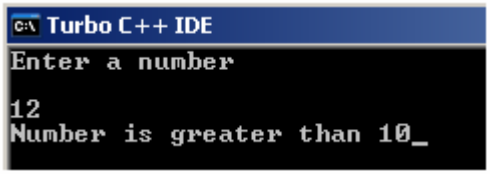
فگر نمبر 9

اچو ته هاڻي هڪ مثال وٺون ته پروگرامنگ ۾ اِف جو استعمال ڪيئن ڪجي.

پروگرام نمبر 30

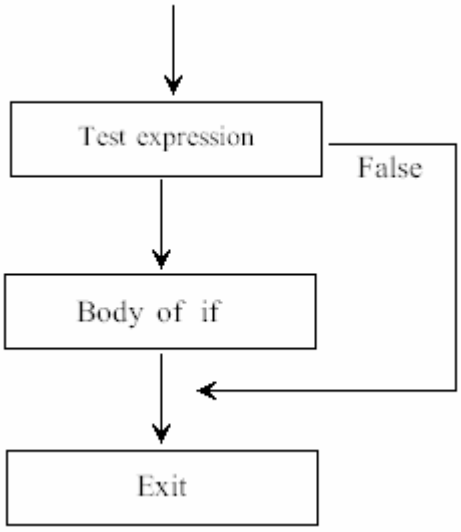
```
#include<constream.h>
void main()
{
clrscr();
int a;
cout<<"Enter a number"<<endl;
cin>>a;
if(a>10)
cout<<"Number is greater than 10";
getch();
}
```

هن پروگرام جو آئوٽ پٽ ڪجهه هن ريت نڪرندو.



اچو ته معلوم ڪريون هن پروگرام جو آئوٽ پٽ ڪيئن نڪتو.

سڀ کان پهريان هڪ انٽيجر ويري ايبل a ڊڪليئر ڪيل آهي، جنهن ۾ اسان سي اين ذريعي ويليو وٺنداسين. پوءِ if(a>10) جيڪڏهن ويري ايبل a جي ڏنل ويليو 10 کان مٿي هجي ته پوءِ سي آئوٽ وارو اسٽيٽمنٽ پرنٽ ڪرائي ۽ جي a جي ويليو گهٽ يا برابر هجي 10 جي ته پروگرام ختم ٿي وڃي.



فگر نمبر 10

اوهان کي فگر نمبر 10 ۾ اِف جو سڄو ڪم سمجهايل آهي.

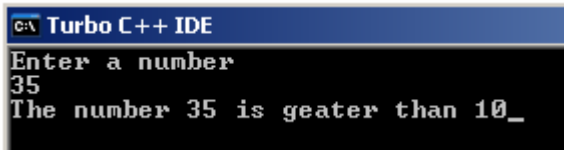
ملٽي پل اسٽيٽمنٽ Multipule statement:

اچو ته هاڻي اِف جي بادي ۾ هڪ کان وڌيڪ اسٽيٽمنٽ پرنٽ ڪرائڻ سکون.

پروگرام نمبر31

```
#include<constream.h>
void main()
{
clrscr();
int a;
cout<<"Enter a number"<<endl;
cin>>a;
if(a>10)
{
cout<<"The number "<<a;
cout<<" is geater than 10";
}
getch();
}
```

هن پروگرام جو آئوٽ پٽ ڪجهه هن ريت نڪرندو.



هن پروگرام جو آئوٽ پٽ اوهان کي اڳئي پروگرام نمبر 30 ۾ سمجهايل آهي، هن پروگرام ۾ صرف هڪ اسٽيٽمنٽ وڌيڪ پرنٽ ڪرايل آهي انهي لاءِ ٻہ بريسز استعمال ڪيل آهن.

نيسٽيڊ اِف انسائيڊ لوپ Nested if inside loop:

اسان مٿي نيسٽيڊ فار لوپ جي باري ۾ سکي آيا آهيون. هڪ کان وڌيڪ فار لوپ کي نيسٽيڊ فار لوپ چئبو آهي. ائين هڪ کان وڌيڪ اِف کي نيسٽيڊ اِف چئبو آهي. نيسٽيڊ اِف انسائيڊ لوپ جي معني آهي ته هڪ کان وڌيڪ اِف لوپ جي بادي ۾.

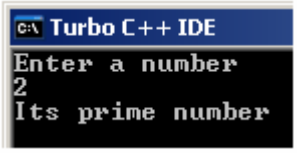
اچو ته نيسٽيڊ اِف انسائيڊ لوپ جو هڪ مثال ونون.

پروگرام نمبر 32

```
#include<constream.h>
#include<process.h>
void main()
{
clrscr();
int n,j;
cout<<"Enter a number"<<endl;
cin>>n;
for(j=2; j<=n/2; j++)
if(n%j==0)
{
cout<<"its not prime number; divisible by"<<j<<endl;
exit(0);
}
cout<<"Its prime number";
```

```
getch();
}
```

هن پروگرام جو آئوٽ پٽ هن طرح نڪرندو.



هن پروگرام ۾ فار لوپ جي اندر ٽيسٽ ايڪسپريشن جي اندر هڪ ٻئي طريقي سان سان ٽيسٽ ايڪسپريش ڏنل آهي. ڪي بورڊ مان ڏنل ويليو کي 2 سان ونڊ ڪيو ويندو، ۽ پوءِ ٽيسٽ ايڪسپريشن جي ڏانهن ڪمپائيلر ويندو. پوءِ وري ڪي بورڊ مان ڏنل ويليو کي ويري ايبل j جي ويليو سان ونڊ ڪيو ويندو جي نمبر پرائيم آيو جھڙوڪ 2 ته پوءِ آخري استيٽمنٽ Its prime number ۽ جي 3 يا ٻيو ڪيو نمبر آيو ته پوءِ its not prime number; divisible by پرنٽ ٿيندو. ۽ جلدئي پروگرام ختم ٿي ويندو چو ته هتي Exit فنڪشن استعمال ٿيل آهي.

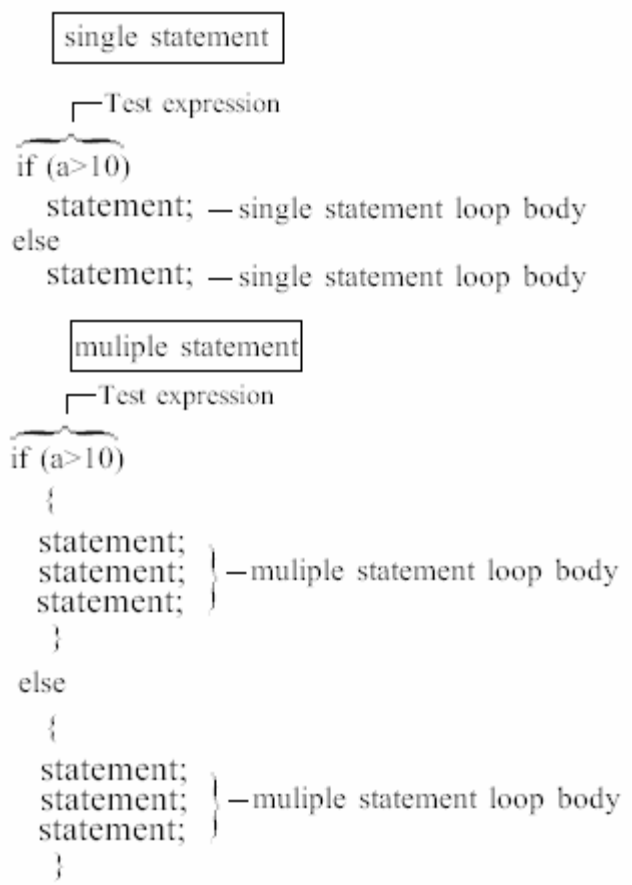
ايگزيٽ Exit ؛

ڪمپائيلر جئين ئي ڏسندو ته نمبر پرائيم ناهي ته هو جلد ئي پروگرام کي ختم ڪري ڇڏيندو چو ته اتي ايگزٽ فنڪشن استعمال ٿيل آهي.
ايگزٽ فنڪشن جو ڪو به ريٽرن ٽائيپ ناهي هتي صرف 0 ارگومنٽ استعمال ٿيل آهي چو ته هي بند ٿيڻ شرط آپريٽنگ سسٽم ڏانهن واپس ويندو. اڪثر ڪري 0 جي جڳهه تي ٻيو نمبر ڏيڻ سان پروگرام ۾ ايرر ايندو آهي.
ايگزيٽ لاءِ اسان هڪ الڳ هيڈر فائيل process استعمال ڪنداسون.

اِف ايلس If else؛

ايلس انگلش ٻولي جو لفظ آهي جنهن جي معنيٰ آهي، ٻيو.
اسان پروگرام نمبر31 ۾ ڏسي آيا آهيون جي ويليو 10 کان مٿي هجي ته استيٽمنٽ پرنٽ ٿئي، پر جي ڪي بورڊ مان ڏنل ويليو 10 کان گهٽ هجي ته پروگرام ختم ٿي وڃي.
پر اِف ايلس استيٽمنٽ ۾ جي هڪ ويليو صحيح آهي ته هڪ استيٽمنٽ پرنٽ ڪرائي. پر جي ٽيسٹ ايڪسپريش غلط ڏنل آهي ته ٻيو استيٽمنٽ پرنٽ ڪرائي.

اچو ته اِف ايلس جو سينٽيڪس لکڻ سکون.



فگر نمبر 11

فگر نمبر 11 ۾ اوهان کي اِف ايلس اسٽيٽمنٽ لکڻ جو طريقو ٻڌايل آهي.

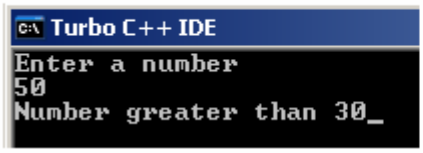
هاڻي اچو تہ اِف ايلس اسٽيٽمنٽ جو هڪ مثال وٺون.

پروگرام نمبر 33

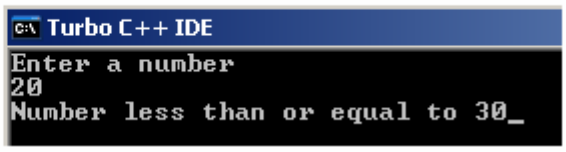
```
#include<constream.h>
void main()
{
clrscr();
int a;
cout<<"Enter a number"<<endl;
cin>>a;
if(a>30)
cout<<"Number greater than 30";
else
cout<<"Number less than or equal to 30";
getch();
}
```

هاڻي اچو تہ هن پروگرام جو آئوٽ پٽ ڏسون.

آئوٽ پٽ 1



آئوٽ پٽ 2



هن پروگرام ۾ هڪ انٽيجر ويري ايبل a مقرر ڪيل آهي، سي اين ذريعي ان انٽيجر ۾ ويليو ورتل آهي. اِف استيٽمنٽ ۾ ڏيکاريل آهي ته ڪي بورڊ مان ڏنل ويليو 30 کان مٿي هجي ته Number greater than 30 پرنٽ ٿئي ۽ ڪي بورڊ مان ڏنل نمبر 30 کان گهٽ هجي ته number less than or equal to 30 پرنٽ ٿئي.

گيٽ سي ايچ اِي getche() :

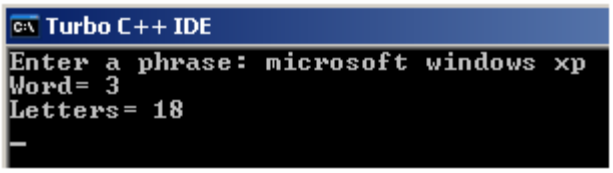
اسان گيٽ سي ايچ متعلق پڙهي آيا آهيون ته هي پروگرام ڪي اسڪرين تي بيهاريندو آهي ته جيئن يوزر آئوٽ پُٽ صحيح ريت ڏسي سگهي.

پر گيٽ سي ايچ اِي getche ڪي اسان بغير ڪنهن انٽر پريس ڪرڻ جي پنهنجو ڪم ڏيکاري ٿي. ڏسو مثال:

پروگرام نمبر 34

```
#include<constream.h>
void main()
{
clrscr();
int chcount=0;
int wdcount=1;
char ch='a';
cout<<"Enter a phrase: ";
while (ch!='\r')
{
ch=getche();
if(ch==' ')
wdcount++;
else
chcount++;
}
cout<<"\nWord= "<<wdcount<<endl;
cout<<"Letters= "<<(chcount-1)<<endl;
getch();
}
```

هن پروگرام جو آئوٽ پُٽ ڪجهه هن ريت نڪرندو.



هن پروگرام جو ڪم آهي ته اسان جيڪي لفظ ۾ اکر لکون انهن کي ڳڻي آخر ۾ اسان کي رزلٽ ڏئي. جيئن اسان Microsoft windows xp لکيو آهي ته اسان جڏهن انٽر پريس ڪيو ته اسان کي خبر پئجي وئي ته اسان 18 اکر ۽ 3 لفظ استعمال ڪيا آهن.

اسان اڳئي پروگرامن ۾ سي اين استعمال ڪندا هئاسين جنهن ۾ اسان ويليو پُٽ ڪري انٽر پريس ڪندا هئاسين ته پوءِ اسان کي آئوٽ پٽ ملندو هو. پر پروگرام نمبر 34 ۾ اسان هڪ اکر لکي وري انٽر پريس ڪيون ته اهو ڪافي مشڪل ۽ عجيب لڳندو. گيٽ سي ايچ اِي فنڪشن ذريعي اسان جيڪو ڪجهه لکنداسون اهو بغير ڪنهن انٽر پريس ڪرڻ جي ڪريڪٽر ويري ايبل ۾ سيو ڪندو.

هن پروگرام ۾ سي کان پهريان به انٽيجر ويري ايبل chcount ۽ wdcount مقرر ڪيل آهن، wdcount جي ويليو 1 مقرر ڪيل آهي. هڪ ڪريڪٽر ويري ايبل ch مقرر ڪيل آهي. وائيل لوپ ۾ ڏيکاريل آهي ته while(ch!='\r') جيڪڏهن يوز انٽر پريس ڪري نه ڪري ته لوپ هلندو رهي. رهي پر جي يوزر انٽر پريس ڪري ته پوءِ لوپ ختم ٿي ۽ آخري ۾ استيٽمنٽ پرنٽ ڪرائي.

If (ch= = ' ') پوءِ ٻيو وائيل لوپ استعمال ٿيل آهي جنهن ۾ ٻڌايل آهي جيڪڏهن ڪريڪٽر ويري ايبل ch ۾ خالي جڳهه يعني اسپيس ڏنو وڃي ته پوءِ wdcount + + ۾ هڪ جو اضافو ٿئي جيئن اسان Microsoft ليڪ اسپيس ڏنو آهي ته پوءِ wdcount++ جي ويليو 2 ٿي وئي، چو ته اسان اڳ ۾ ئي wdcount جي ويليو 1 مقرر ڪيا آهيون، هاڻي اهو هڪ مان وڌي 2 ٿي ويو آهي. پوءِ اسان windows لکي اسپيس ڏنو ته wdcount جي ويليو 3 ٿي وئي آخر ۾ اسان xp لکيو آهي اسان xp لکي اسپيس ڏيڻ جي بجاءِ انٽر ڪيو آهي انهي لاءِ wdcount جي ويليو ساڳي 3 آهي جيڪا آخر ۾ پرنٽ ڪرايل آهي.

ٻئي وري chcount آهي وائيل لوپ ۾ چيل آهي ته جڏهن اسان اسپيس ڏيون ته wdcount جي ويليو وڌي پر جي اسان اسپيس نه ڏيون ته پوءِ chcount جي ويليو وڌي.

ايسڪيپ سيڪيونس Escape sequence:

هن پروگرام ۾ اوهان کي ٻہ مختلف ڪوڊ '\r' ۽ '\n' نظر اچي رهيا آهن، انهن کي ايسڪيپ سيڪيونس چيو وڃي ٿو.

سي پلس پلس ۾ ٽوٽل 10 ايسڪيپ سيڪيونس آهن، انهن جا نالا ۽ ڪم هيٺ ڏجن ٿا.

ايسڪيپ سيڪيونس	نالو	ڪم
\a	بيپ	پروگرام جي شروع ۾ آواز اچڻ
\b	بيڪ اسپيس	هي هڪ اسپيس کي پوئتي ڪندو
\f	فارم فيڊ	♀ کي اسڪرين ۾ پرنٽ ڪرائڻ
\n	نيولائين	نئين لائين ۾ اچڻ، ايندڙ وارو ڪم
\r	ريٽرن	انٽر لاءِ استعمال ٿيندو آهي
\t	ٽيب	اٺ ڪريڪٽر ڇڏڻ لاءِ
\\	بيڪ سليش	بيڪ سليش پرنٽ ڪرائڻ لاءِ
\'	سنگل ڪوٽيشن مارڪ	سنگل ڪوٽيشن پرنٽ ڪرائڻ لاءِ
\"	ڊبل ڪوٽيشن مارڪ	ڊبل ڪوٽيشن پرنٽ ڪرائڻ لاءِ
\xdd	هيگاڊيسيمل نوٽيشن	هيگاڊيسيمل نوٽيشن پرنٽ ڪرائڻ لاءِ

اچو تہ ڏسون تہ ايسڪيپ سيڪيونس کي پروگرام ۾ ڪيئن استعمال ڪجي.

پروگرام نمبر 35

```
#include<constream.h>
void main()
{
clrscr();
cout<<"Information Technology \a";
getch();
}
```

اوهان کي جتي \a نظر اچي رهيو آهي اوهان انهي کي پروگرام ۾ ڪٽي مختلف ايسڪيپ سيڪيونس استعمال ڪري انهن جو ڪم ڏسي سگهو ٿا. هن پروگرام جي شروعات ۾ هڪ آواز ايندو، چو تہ هن پروگرام ۾ \a استعمال ٿيل آهي

نيسٽيڊ اِف Nested if:

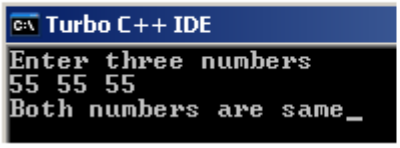
هڪ کان وڌيڪ اِف جو استعمال ڪرڻ:

پروگرام نمبر 36

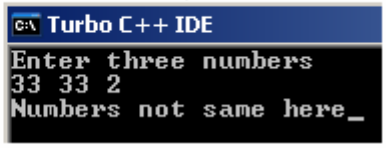
```
#include<constream.h>
void main()
{
clrscr();
int a,b,c;
cout<<"Enter three numbers: use space B/W num"<<endl;
cin>>a>>b>>c;
if(a==b)
if(b==c)
cout<<"Both numbers are same";
else
cout<<"Numbers not same here";
getch();
}
```

هن پروگرام جو آئوٽ پُٽ ڪجهه هن ريت نڪرندو. پروگرام نمبر 36 جو ڪم پڙ مٿي ڏيکاريل پروگرامن جهڙو آهي، هن ۾ صرف هڪ اِف جو وڌيڪ استعمال ٿيل آهي.

آئوٽ پُٽ 1



آئوٽ پُٽ 2



اِلس ... اِف ڪنسٽرڪشن else is constraction:

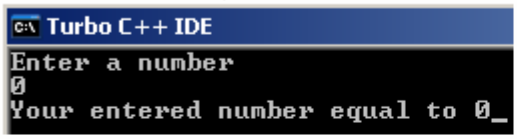
اسان اڳئي هڪ اِلس استعمال ڪندا هئاسين پر هاڻي اسان هڪ کان وڌيڪ اِلس استعمال ڪنداسون.

پروگرام نمبر 37

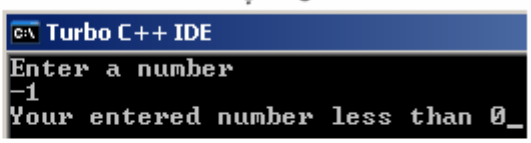
```
#include<constream.h>
void main()
{
clrscr();
int test;
cout<<"Enter a number"<<endl;
cin>>test;
if(test>0)
cout<<"Your entered number greater than 0";
else if(test<0)
cout<<"Your entered number less than 0";
else if(test==0)
cout<<"Your entered number equal to 0";
getch();
}
```

هن پروگرام جو آئوٽ پٽ ڪجھ هن ريت نڪرندو.

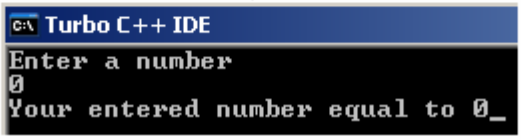
آئوٽ پُٽ 1



آئوٽ پُٽ 2



آئوٽ پُٽ 3



هن پروگرام ۾ هڪ ڪان وڌيڪ اِلس ۽ اِف جو استعمال ٿيل آهي. اسان جو انٽر ڪيل نمبر جنهن ۾ ٽيسٽ ايڪسپريشن ۾ صحيح آهي اهو وري پنهنجي هيٺيان ڏنل اسٽيٽمنٽ پرنٽ ڪندو ۽ جي ٽيسٽ ايڪپريشن غلط هجي تہ پوءِ هو ٻئي ٽيسٽ ايڪسپريشن ڏانهن ڪمپائيلر کي موڪليندو.

سوئيچ اسٽيٽمنٽ Switch statement:

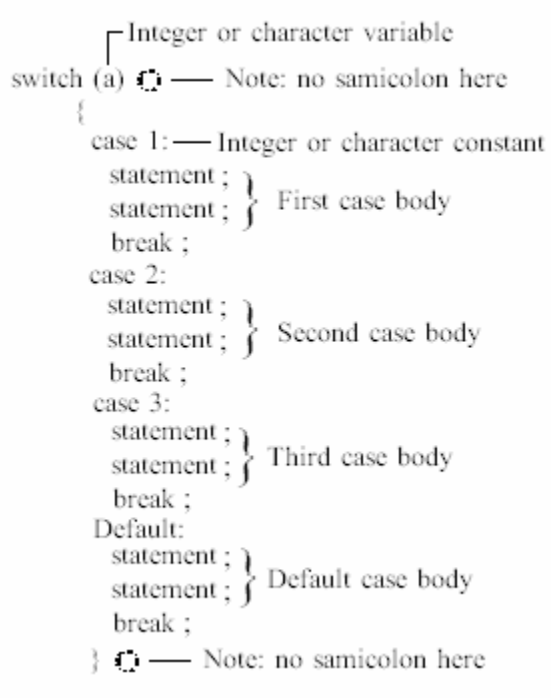
اِلس ... اِف: اسان پروگرام نمبر 37 ۾ ڏسي آيا آهيون تہ هڪ ڪان وڌيڪ ڪيئن ويليو پرنٽ ڪرائجن، پر اسان صرف هڪ ان پُٽ ڏنو جيڪو ۾ ٽيسٽ ايڪسپريشن صحيح هجي اهو اسٽيٽمنٽ پرنٽ ٿئي.

سوئيچ اسٽيٽمنٽ: هن ۾ يوزر کي هڪ ڪان وڌيڪ چوائز ڪرڻ جو اختيار آهي، هو پنهنجي مرضي مطابق ڪو ۾ اسٽيٽمنٽ پرنٽ ڪرائي سگهي ٿو. سوئيچ اسٽيٽمنٽ هڪ وڻ جيان اوهان جي دل جنهن طرف چئي ان طرف چڙهي سگهو ٿا.

اوهان عام سافٽويئر ۾ ڏٺا هوندا جن ۾ اوهان کي هڪ ڪان وڌيڪ چوائز ڏنل هوندا آهن، جن مان اوهان کي صرف هڪ جي چوائز ڪرڻي پوندي آهي.

ساڳئي طرح سوئيچ جي ذريعي اوهان اهڙو ڪجھ ناهي سگهو ٿا.

اچو ته سوئيچ جو سينٽيڪس لکڻ جو طريقو ڏسون.



فگر نمبر 12

اوهان کي مٿي ڏيکاريل آهي ته ڪيئن سوئيچ استيٽمنٽ ڪم ڪندو آهي، اوهان جي انٽر ڪيل آئوٽ پُٽ کي هر ڪيس سان ميچ ڪندو جيڪو به استيٽمنٽ صحيح هوندو ان کي پرنٽ ڪندو. پر اوهان جو انٽر ڪيل ان پٽ ڪنهن به ڪيس سان ميچ نہ ٿو ڪري ته پوءِ آخر ۾ ڈنل ڊفالٽ وارو استيٽمنٽ پرنٽ ڪندو.

اچو ته سوئيچ جو هڪ مثال ونون.

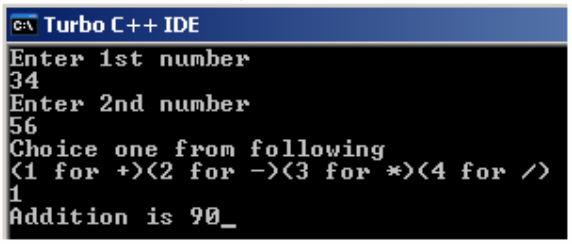
پروگرام نمبر 38

```
#include<constream.h>
void main()
{
clrscr();
int a,b;
int choice;
cout<<"Enter 1st number"<<endl;
cin>>a;
cout<<"Enter 2nd number"<<endl;
cin>>b;
cout<<"Choice one from following"<<endl;
cout<<"(1 for +)(2 for -)(3 for *)(4 for /)"<<endl;
cin>>choice;
switch(choice)
{
case 1:
cout<<"Addition is "<<a+b;
break;

case 2:
cout<<"Subtaction is "<<a-b;
```

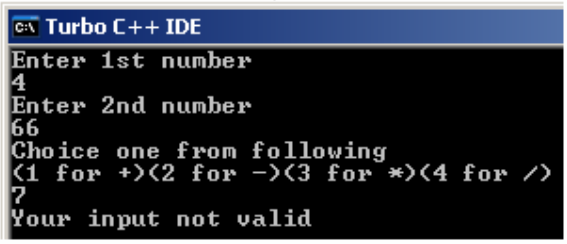
```
break;

case 3:
cout<<"Multiplication is "<<a*b;
break;

case 4:
cout<<"Division is "<<a/b;
break;

default:
cout<<"Your input not valid";
}
getch();
}
```

هن پروگرام جو آئوٽ پُٽ ڪجهه هن ريت نڪرندو.

آئوٽ پُٽ 1



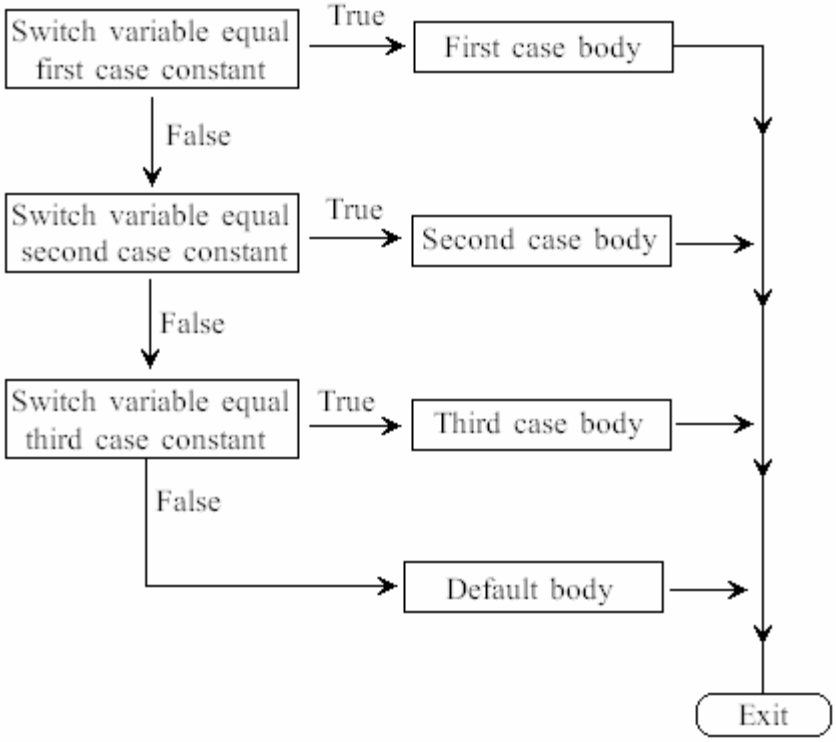
آئوٽ پُٽ 2



آئوٽ پُٽ 1: هن آئوٽ پُٽ ۾ ٻه انگ 56 , 34 ڏنل انهن جي هينيان اوهان لاءِ 4 چوائز ڏنل آهن، جنهن ۾ اسان 1 يعني + کي سڪليٽ ڪيو آهي لاءِ 34، 56 جو جوڙ ڏيکاريل آهي.

آئوٽ پُٽ 2: هن ۾ اسان ٻه انگ ڏنا آهن، پر اسان آخر ۾ چوائز ۾ غلط چوائز يعني 7 انٽر ڪيو آهي. جنهن جي ڪري پروگرام جو آئوٽ Your input not valid آيو آهي.

هن پروگرام ۾ ٽي انٽيجر ويري ايبل a , b , choice ڊڪليئر ڪيل آهن. اسان پهرين ٻه ويري ايبلز يعني a ۽ b ۾ اسان ٻه ويليو جوڙ ڪٽ وغيره لاءِ ورتا آهن. ٽيئن ويري ايبل کي اسان جوڙ ڪٽ وغيره جي چوائز لاءِ ڪتب آندو آهي. اسان سي اين ذريعي choice ويري ايبل ۾ ويليو وٺنداسين. اسان سوئيچ ۾ ان ويري ايبل کي رکيو آهي. اسان جڏهن choice ويري ايبل ۾ ويليو 1 ڏينداسين ته هو case 1 کي پرنٽ ڪندو جنهن ۾ جوڙ لاءِ استيٽمنٽ نهيل آهي. اسان جڏهن choice ويري ايبل ۾ 2 ويليو ڏينداسين ته هو case 2 ڏانهن ويندو جنهن ۾ ڪٽ لاءِ استيٽمنٽ لکيل آهي. اوهان متي آئوٽ پُٽ 2 تي غور ڪيو جنهن ۾ Your input not valid پرنٽ ٿيل آهي جو ته اسان 7 اِن پُٽ ڏنو آهي جيڪو ڪنهن به case سان ميچ نه ٿو ڪري انهي لاءِ آخري استيٽمنٽ جيڪو ڊيفالٽ ۾ ڏنل آهي اهو پرنٽ ٿيل آهي.



فگر نمبر 13

فگر نمبر 13 ۾ اوهان کي سوئيچ استيٽمنٽ جو سڄو ڪم سمجهايل آهي. اوهان هن کي ڏسي سوئيچ استيٽمنٽ کي آساني سان سمجهي سگهو ٿا.

بريڪ استيٽمنٽ Break statement :

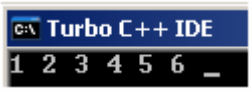
اوهان کي پروگرام نمبر 38 ۾ بريڪ استيٽمنٽ نظر اچي رهيو آهي، هتي بريڪ جو ڪم آهي ته هو ڪمپائيلر کي لوپ (پروگرام) مان ٻاهر ڪڍي.

اچو ته بريڪ استيٽمنٽ جو هڪ خالص مثال ونون.

پروگرام نمبر 39

```
#include<constream.h>
void main()
{
clrscr();
for(int a=1; a<=10; a++)
{
if(a==7)
break;
cout<<a<<" ";
}
getch();
}
```

پروگرام نمبر 39 جو آئوٽ پٽ ڪجھ هن ريت نڪرندو.



اوهان ڏسو فار لوپ ڏانهن جنهن کي 10 چڪر هلڻو هو پر هو صرف 6 چڪر هليو ۽ 7 چڪر تي بريڪ استيٽمنٽ ان کي بيهاري ڪمپائيلر کي لوپ (پروگرام) مان ڪڍي ڇڏيو.

ڪنٽينيو استيٽمنٽ continue statement؛

ڪنٽينيو انگلش ٻولي جو لفظ آهي جنهن جي معنيٰ آهي جاري رکڻ. سي پلس پلس ۾ به هن جو ڪم ساڳيو آهي. هي ڪمپائيلر کي هڪ مقرر ڪيل جڳھ کان اڳتي موڪلينڈو آهي.
اچو ته ڪنٽينيو استيٽمنٽ جو مثال وٺون.

پروگرام نمبر 40

```
#include<constream.h>
void main()
{
clrscr();
for(int a=1; a<=10; a++)
{
if(a==7)
continue;
cout<<a<<" ";
}
getch();
}
```

هن پروگرام جو آئوٽ پُٽ ڪجھ هن ريت نڪرندو.



مٿي آئوٽ پٽ ۾ اوهان کي 7 نظر نہ ايندو چو ته اسان اتي ڪنٽينيو استيٽمنٽ جو استعمال ڪيو آهي، جيڪو ڪمپائيلر کي اتان اڳتي موڪلينڈو.

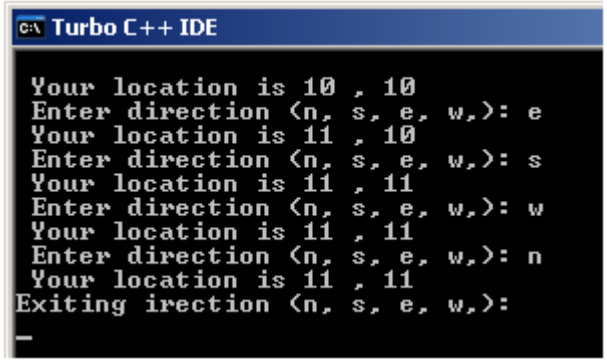
سوئيچ استيٽمنٽ وِٿ ڪريڪٽر ويري ايبل Sw: with Char varble ؛

اسان سوئيچ استيٽمنٽ کي پروگرام نمبر 38 ۾ انٽيجر ويري ايبل سان استعمال ڪري آيا آهيون، هاڻي اسان سوئيچ استيٽمنٽ کي ڪريڪٽر ويري ايبلز سان استعمال ڪنداسين.
اچو ته سڀ کان پهريان هڪ مثال وٺون ۽ پوءِ ان مثال کي ڏسي سوئيچ استيٽمنٽ وِٿ ڪريڪٽر ويري ايبل تي وڌيڪ ڳالهيون.

پروگرام نمبر 41

```
#include<constream.h>
void main()
{
clrscr();
textmode(3);
char dir='a';
int x=10, y=10;
while(dir != '\r')
{
cout<<"\n Your location is "<<x<<" , "<<y;
cout<<"\n Enter direction (n, s, e, w,): ";
dir=getche();
switch(dir)
{
case 'n':
y;
break;

case 's':
y++;
break;

case 'e':
x++;
break;

case 'w':
x;
break;

case '\r':
cout<<"Exiting \n";
break;

default:
cout<<"Try again\n";
}
}
getch();
}
```

هن پروگرام جو آئوٽ پُٽ ڪجهه هن ريت نڪرندو.



اچو ته ڏسون هن پروگرام جو آئوٽ پُٽ ڪيئن آيو.

اوهان کي چار ڪريڪٽر n, s, e, w, ڏنل آهن اوهان کي انهن مان ڪو به هڪ چوائز ڪرڻو آهي. جڏهن اوهان s پريس ڪندو ته y جي ويليو 10 مان 11 ٿي ويندي ۽ اوهان وري s پريس ڪندو ته y جي ويليو 12 ٿي ويندي.
وري اوهان e پريس ڪندو ته x جي ويليو 10 مان 11 ٿي ويندي.
اسان هن پروگرام ۾ به ويري ايبل x ۽ y ڊڪليئر ڪري انهن جي ويليو 10 مقرر ڪئي. اوهان مٿي پروگرام جي سوئيچ ڪنديشن ڏانهن ڏسو جڏهن اسان w ۽ n پريس ڪنداسين ته x ۽ y جي ويليوز ۾ ڪو به فرق نه ايندو.
پر اسان جيترا دفعا e پريس ڪندا سين ته اوترا چڪر x جي ويليو ۾ اضافو ٿيندو. ۽ جيترا دفعا s پريس ڪنداسين ته اوترا دفعا y جي ويليو ۾ هڪ هڪ جو اضافو ٿيندو.

ڊيفالٽ Defaulat :

هي ڪي ورڊ سوئيچ اسٽيٽمنٽ جي آخر ۾ رکبو آهي ته جيئن يوزر مقرر ڪيل ويليو جي جاءِ تي بي ڪا سليڪشن ڪري ته ڊيفالٽ وارو اسٽيٽمنٽ پرنٽ ٿئي مثال:

Error: Invalid selection وغيره.

ڊيفالٽ اسٽيٽمنٽ يوزر کي آگاهي ڏيڻ لاءِ استعمال ٿيندو آهي، ته هو ڪجهه غط ڪري رهيو آهي جيڪو مطلوبہ پروگرام ۾ جائز ناهي.
ڊيفالٽ وارو اسٽيٽمنٽ پرنٽ ٿيڻ شرط ڪنهن به صورت ۾ سوئيچ اسٽيٽمنٽ کي ختم ٿيڻو پوندو.

لاجيڪل آپريٽر Logical operator:

اسان اڳئي ٻن قسمن جا آپريٽر يعني ارٿ ميٽڪ % /, *, -, +, ۽ رليشن آپريٽر >=, <=, >, <, != ,== بابت پڙهي آيا آهيون.
هاڻي اسان آپريٽر جي ٽيئن فيملي يعني لاجيڪل آپريٽر جي باري ۾ پڙهنداسين.
سي پلس پلس ۾ ٻن قسمن جا لاجيڪل آپريٽر آهن.

لاجيڪل آپريٽر	نالا
&&	ايند آپريٽر and operator
||	آر آپريٽر or operator

هڪ مثال ونو: جڏهن بل گيٽس وري مائيڪرو سافٽ جوائين ڪندو ۽ مائيڪرو سافٽ وسٽا کان وڌيڪ ٻيو آپريٽنگ سسٽم ناهيندو ته ان وقت اسان به ڪمپيوٽر وٺنداسين.
هن مثال ۾ به هاڪاري ڳالهيون چيل آهن ۽ اسان جو ري ايڪشن به هاڪاري آهي.

پهرين لاجيڪل آپريٽر يعني and operator ۾ ٻئي ڪنديشن صحيح هجن ته اسٽيٽمنٽ پرنٽ ڪندو، نه ته وري ٻيو ڪنديشن جيڪو اِلس ۾ ڏنل هوندو اهو پرنٽ ڪندو.

پهريون ڪنڊيشن	ٻيو ڪنڊيشن	نتيجو
صحيح	صحيح	صحيح
غلط	صحيح	غلط
صحيح	غلط	غلط

ٻيو مثال: بل گيٽس مائيڪرو سافٽ جوائين ڪري يا نہ پر جڏهن بہ مائيڪرو سافٽ وسٽا کان وڌيڪ آپريٽنگ سسٽم متعارف ڪرايو تہ اسان ڪمپيوٽر ونڊاسين.

هن مثال ۾ هڪ ناڪاري ۽ ٻيو هاڪاري ڳالهہ چيل آهي، پر اسان جو ري ايڪشن وري بہ هاڪاري آهي.

ٻيو لاجيڪل آپريٽر يعني or operator بہ مثال ۾ جهڙو ڪم ڪري ٿو هڪ ڪنڊيشن (هڪ ڳالهہ هاڪاري) صحيح هجي تہ استيٽمنٽ پرنٽ ڪرائي.

پهريون ڪنڊيشن	ٻيو ڪنڊيشن	نتيجو
صحيح	صحيح	صحيح
غلط	صحيح	صحيح
صحيح	غلط	صحيح

هاڻي اچو تہ هنن لاجيڪل آپريٽرز جا مثال ونون.

اينڊ آپريٽر:

اينڊ آپريٽر ۾ ٻئي ڪنڊيشن صحيح هجڻ گهرجن. تہ جيئن اسان جو مقرر ڪيل استيٽمنٽ پرنٽ ٿئي.

اچو تہ اينڊ آپريٽر جو هڪ مثال ونون.

پروگرام نمبر 42

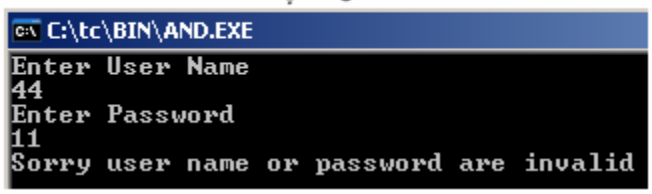
```
#include<constream.h>
void main()
{
clrscr();
int a,b;
cout<<"Enter User Name"<<endl;
cin>>a;
cout<<"Enter Password"<<endl;
cin>>b;
if(a==23 && b==11)
cout<<"Welcome to microsoft windows";
else
cout<<"Sorry user name or password are invalid";
getch();
}
```

پروگرام نمبر 42 جو آئوٽ پُٽ ڪجھ هن ريت نڪرندو.

آئوٽ پُٽ 1



آئوٽ پُٽ 2



اوهان مٿي پروگرام نمبر 42 ۾ ڏسو اتي ويري ايبل a جي ويليو 23 ۽ ويري ايبل b جي ويليو 11 مقرر ڪيل آهي. جڏهن اسان آئوٽ پُٽ 1 ۾ ٻئي ويليو صحيح ڏنل آهن يعني 23 ۽ 11 انهي لاءِ اسان جو پهريون اسٽيٽمنٽ پرنٽ ٿيل آهي. پر ٻئي آئوٽ پُٽ ۾ هڪ ويليو 23 جي جڳهه تي 44 ڏنل آهي انهي لاءِ اسان آخري اسٽيٽمنٽ پرنٽ ٿيل آهي.

آر آپريٽر Or operator :

هن آپريٽر ۾ بن مان هڪ ڪنڊيشن صحيح هجڻ گهرجي. ٻئي اسٽيٽمنٽ غلط هجڻ ڪري پروگرام غلط ٿي ويندو.

اچو ته آر آپريٽر جو هڪ مثال ونون.

پروگرام نمبر 43

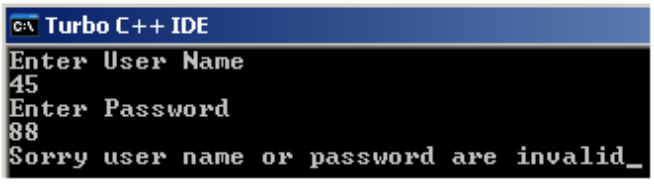
```
#include<constream.h>
void main()
{
clrscr();
int a,b;
cout<<"Enter User Name"<<endl;
cin>>a;
cout<<"Enter Password"<<endl;
cin>>b;
if(a==23 || b==11)
cout<<"Welcome to microsoft windows";
else
cout<<"Sorry user name or password are invalid";
getch();
}
```

هن پروگرام جو آئوٽ پُٽ ڪجھ هن ريت نڪرندو.

آئوٽ پُٽ 1



آئوٽ پُٽ 2



توهان آئوٽ پُٽ 1 ۾ ڏسو ته 23 جي جڳهه تي 44 ڏنل آهي پر جيئن ته هڪ ڪنڊيشن يعني 11 صحيح آهي انهي لاءِ پهرين ڪنڊيشن پرنٽ ٿيل آهي. ٻئي آئوٽ پٽ ۾ ٻئي ڪنڊشن غلط ڏنل آهن انهي لاءِ ٻيو نمبر اسٽيٽمنٽ پرنٽ ٿيل آهي.

گو ٽو استيٽمنٽgoto satatement؛

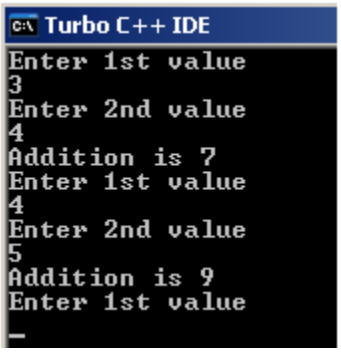
هن جي ذريعي اوهان ڪمپائيلر کي ڪنهن خاص جڳهه تہ موڪلي سگهو ٿا. پر اڪثر سي پلس پلس جي پروگرامن ۾ هن جو استعمال گهٽ ٿيندو آهي، پر تنهن هوندي بہ هي سي پلس پلس جو حصو آهي هن جي باري ۾ سکڻ ضروري آهي.

اچو تہ گو ٽو جو هڪ مثال وٺون.

پروگرام نمبر 44

```
#include<iostream.h>
#include<conio.h>
int main()
{
clrscr();
int a,b;
restart:
cout<<"Enter 1st value"<<endl;
cin>>a;
cout<<"Enter 2nd value"<<endl;
cin>>b;
cout<<"Addition is "<<a+b<<endl;
goto restart;
getch();
return 0;
}
```

هن پروگرام جو آئوٽ پٽ ڪجهه هن ريت نڪرندو.

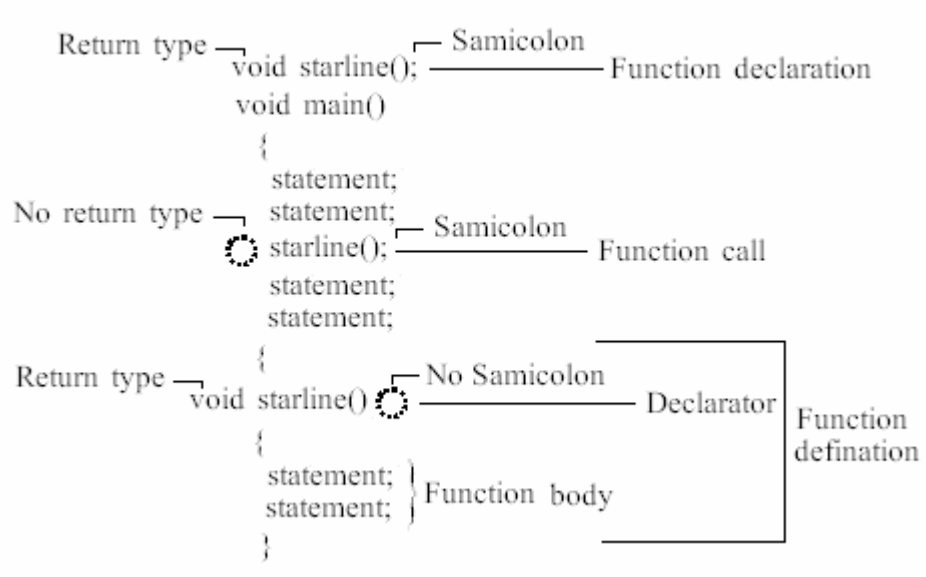


هن پروگرام ۾ اسان جڏهن هن پروگرام ۾ بہ ويليوز ڏئي انٽر ڪندا آهيون تہ ڪمپائيلر انهن جو جوڙ ڏيکاري وري بہ ويليو انٽر ڪرڻ لاءِ چوندو آهي تہ Enter 1st value ڏيکاريندو آهي. اوهان پروگرام نمبر 44 جي ڪوڊنگ ڏي ڏسو جتي بہ ويري ايبل ڊڪليئر ڪيل آهن بلڪل انهن جي هيٺيان هڪ ايڊنٽيفائير restart ڏنل آهي. ۽ پروگرام جي آخر ۾ goto restart ڏنل معنيٰ تہ ڪمپائيلر وري restart ايڊنٽيفائير ڏانهن وڃي. ۽ پروگرام وري شروع ٿئي.

باب نائون
فنڪشن

Function

اسان مٿي ذڪر ڪري آيا آهيون ته سي پلس پلس ۾ فڪشن پيڙهه واري پت جي حيثيت رکي ٿو. دراصل هڪ پروگرام گهڻن فنڪشنز جو ميڙ هوندو آهي، وري اهڙي طرح فنڪشن گهڻن اسٽيٽمنٽز جو ميڙ هوندو آهي.
اسان پروگرام جي ڪوڊنگ کي صحيح سمجهڻ ۽ پروگرام جي سائيز گهٽ ڪرڻ لاءِ فنڪشن استعمال ڪندا آهيون.
سي پلس پلس ۾ ٻن قسمن جا فنڪشن آهن هڪ پري ڊفائينڊ ۽ ٻيو يوزر ڊيفائينڊ.
پري ڊيفائينڊ فڪشن سي پلس پلس سان گڏ ايندا آهن، جيئن گيٽ سي ايچ، سي اين ۽ سي آئوٽ وغيره.
يوزر ڊيفائينڊ اِهي فنڪشن آهن جيڪي يورز پاڻ ناهيندو آهي. جن جو مثال اوهان کي هاڻي ڏنو ويندو.
اچو ته فنڪشن جي سينٽيڪس لکڻ سکون.



فگر نمبر14

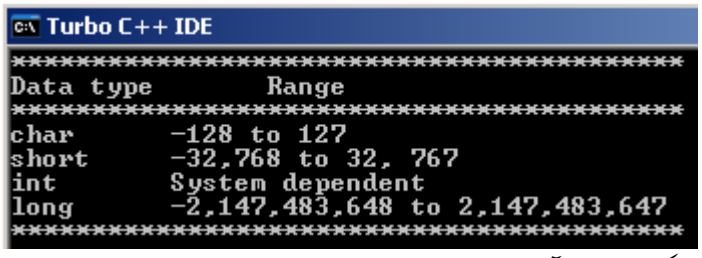
اچو ته هاڻي هڪ سمپل فنڪشن جو مثال وٺون.

پروگرام نمبر 45

```
#include<constream.h>
void starline();
void main()
{
clrscr();
starline();
cout<<"Data type       Range "<<endl;
starline();
cout<<"char      -128 to 127"<<endl;
cout<<"short     -32,768 to 32, 767"<<endl;
cout<<"int       System dependent"<<endl;
cout<<"long      -2,147,483,648 to 2,147,483,647"<<endl;
```

```
starline();
getch();
}
void starline()
{
for(int a=1; a<=42; a++)
cout<<'*';
cout<<endl;
}
```

هن پروگرام جو آئوٽ پٽ ڪجهه جن ريت نڪرندو.



اچو ٿ غور ڪريون ٿ هن پروگرام جو آئوٽ پٽ ڪيئن نڪتو.

اسان پروگرام جي شروع ۾ هڪ فنڪشن void main() استعمال ڪندا هئاسين، پر هن پروگرام ۾ اسان هڪ ٻيو فنڪشن void satarline() ٻڌو آهي جنهن کي فنڪشن ڊڪليئريشن چيو وڃي ٿو، void starline() کي فنڪشن پروٽوٽائيپ ٻ چيو وڃي ٿو. جڏهن اسان هڪ پروٽو ٽائيپ فنڪشن ڊڪليئر ڪندا آهيون ٿ دراصل اسان ڪمپائيلر کي اها آگاهي ڏيندا آهيون ٿ هن پروگرام ۾ اڳتي هڪ فنڪشن آهي، جيڪو پروگرام جي هيٺ ڏنل آهي.

clrscr(); فنڪش کان پوءِ وري starline(); فنڪشن ڏنل آهي، جنهن کي فنڪشن ڪال چيو وڃي ٿو، جتي ٻ اسان starline(); فنڪشن لکنداسين ٿ هو آخر ۾ فنڪشن ڊيڪلريٽر ڏانهن ويندو، جتي هڪ فار لوپ ڏنل جنهن چيل آهي ٿ 42 چڪر استار پرنٽ ڪرايا وڃن.

فنڪشن کي سمجهڻ سولو ڪم آهي جتي ٻ starline(); فنڪشن جو استعمال ڪيو ويندو اهو پرگرام جي آخر ۾ ڊيڪلريٽر ڏانهن ويندو.

فنڪشن ڊڪليئريشنFunction Declaration؛

جيئن اسان ڪنهن ٻ ويري ايبل کي استعمال کان ڪرڻ کان پهريان انهي کي ڊڪليئر ڪندا آهيون بلڪل ساڳئي طرح اسان کي فنڪشن متعلق ڪمپائيلر کي ٻڌائڻو پوندو آهي. ڪمپائيلر کي ٻن طريقن سان فنڪشن متعلق ٻڌايو ويندو آهي.

هڪ ٿ فنڪشن کي پهريان ڊڪليئر ڪري پوءِ ڪمپائيلر کي ٻڌائبو آهي ۽ ٻيو پهريان ڪمپائيلر کي فنڪشن متعلق ٻڌائي پوءِ انکي ڊڪليئر ڪبو آهي.

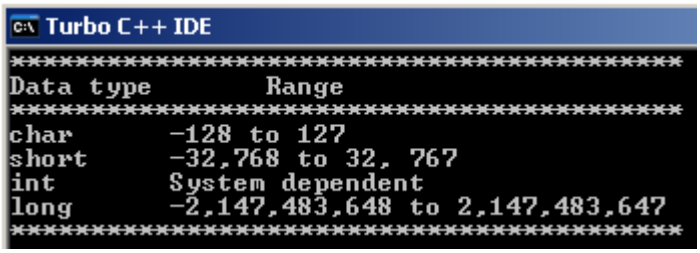
اسان کي پروگرام نمبر 45 ۾ فنڪشن بابت پهريان ڪمپائيلر کي ٻڌايو آهي ٿ starline(); نالي هڪ فنڪشن هن پروگرام ۾ آهي.

هاڻي اسان پهريان فنڪشن کي ڊڪليئر ڪنداسين. اچو ٿ هڪ مثال ونون.

پروگرام نمبر 46

```
#include<constream.h>
void starline()
{
for(int a=1; a<=42; a++)
cout<<'*';
cout<<endl;
}
void main()
{
clrscr();
starline();
cout<<"Data type       Range "<<endl;
starline();
cout<<"char       -128 to 127"<<endl;
cout<<"short      -32,768 to 32, 767"<<endl;
cout<<"int        System dependent"<<endl;
cout<<"long       -2,147,483,648 to 2,147,483,647"<<endl;
starline();
getch();
```

هن پروگرام جو آئوٽ پٽ پروگرام نمبر 45 جھڙو ايندو.



بنهي پروگرام جو ڪم ساڳيو آهي پر هنن ۾ صرف لکڻ يعني سينٽيڪس جو طريقو ٻيو استعمال ٿيل آهي.

ڪالنگ فنڪشن Calling function:

پروگرام نمبر 45 ۽ 46 ۾ اسان ٽي چڪر (ڪالنگ فنڪشن) فنڪشن کي استعمال ڪيو آهي. جيڪو هن ريت نظر اچي رهيو آهي. starline(); .

هميشہ يوزر ڊفائينڊ فنڪشن کي ٻہ پري ڊفائينڊ فنڪشن وانگر پيرنٽيسز يعني بريسز سان بند ڪيو وڃي. اسان جڏهن ٻہ فنڪشن کي ڪال ڪندا آهيون تہ هو فنڪشن ڊيڪلريٽر ڏانهن ويندو ۽ اتان معلومات کڻي اتي پرنٽ ڪرائيندو جتي اسان فنڪشن کي ڪال ڪيو آهي.

فنڪشن ڊيفينيشن Function Defination:

هاڻي اچو تہ فنڪشن جي ڊيفينيشن ڏانهن اچون جيڪو سڃي يوزر ڊفائينڊ فنڪشن جو پيءُ چئي سگهجي ٿو. اسان پروگرام نمبر 45 جي آخر ۾ ۽ پروگرام نمبر 46 جي پهريان جيڪو ٻہ ڪوڊنگ استعمال ڪيو آهي انهي کي فنڪشن ڊيفينيشن چيو وڃي ٿو.

```
void starline()
{
for(int a=1; a<=42; a++)
cout<<'*';
cout<<endl;
}
```

مٿي اوهان کي فنڪشن ڊيفينيشن جي ڪوڊنگ ڏنل آهي.
اچو ته هڪ ٽيبل ذريعي فنڪشن جي ڪامپونيٽ جو مثال ونون.

Component	Purpose	Example
Declaration (prototype)	Specifies function name, argument, types and return values. Alert compiler that a function coming up later.	void starline();
Call	Causes the function to be executed	Starline();

Definition	The function it's self. Contains the lines of codes that constitute the function.	Void starline()
		{
		//lines of code
		}

Decelerator	First line of definition	Void starline()

لائبريري ايند يوزر ڊيفائينڊ فنڪشن:

اسان اڳئي لابريئري فڪشن متعلق پڙهي سکي آيا آهيون اسان ڪجهه هن ريت لائبريري فنڪشن لکندا آهيون.

```
ch=getche();
```

جيئن اسان مٿي سکي آيا آهيون ته يوزر ڊيفائينڊ فنڪشن ۾ شروع يا آخر ۾ انهي جي ڊيفينيشن (فنڪشن جو اصلي حصو) لکڻو پوندو آهي.
اوهان سوچيندا ته لائبريري فنڪشن جو ڊيفينيشن ڪٿي آهي؟ دراصل جڏهن اسان هيڊر فائيل (جيئن getche(); فنڪش لاءِ conio.h) استعمال ڪندا آهيون ته ان فنڪشن جو ڊيفينيشن ان هيڊر فائيل جي اندر لکيل هوندو آهي.

پاسنگ آرگومنٽ ٽو فنڪشن Passing arguments to function:

آرگومنٽ جي اندر ڊيٽا (مثال: انٽيجر) موڪليو ويندو آهي پروگرام کان فنڪشن ڏي. آرگومنٽ اسان کي اجازت ٿو ڏي ته اسان مختلف قسم جا ويليوز فنڪشن جي پرنٽ ڪرايون. يا وري پروگرام جي ضرورت مطابق بيا ڪم ڪيون.

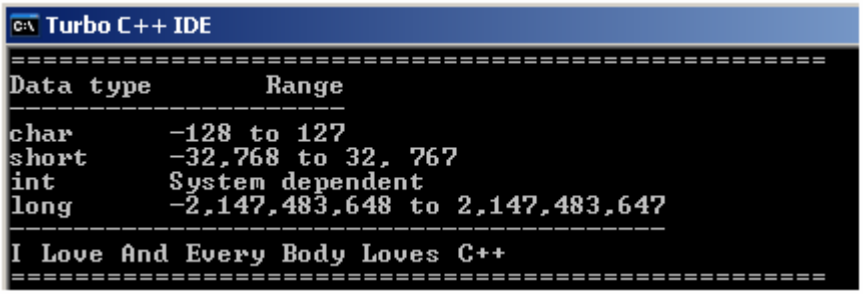
پاسنگ ڪانسٽنس passing constant:

اسان مٿي پروگرام نمبر 45 ۽ 46 ۾ مقرر ڪيل 42 استار پرنٽ ڪرايا، پر اهو عمل وڏن پروگرامز جي اندر هلڻ جوڳو ناهي، ڇو ته وڏن پروگرامز جي هر سٽ جو پنهنجو ضرورت هوندو آهي.
اسان هڪ پروگرام ۾ مختلف ويليوز جا استار يا ٻيو ڪجهه پرنٽ ڪرائي سگهون ٿا، هاڻي اچو ته هڪ مثال وٺون.

پروگرام نمبر 47

```
#include<constream.h>
void repchar(char, int);
void main()
{
clrscr();
repchar('=',50);
cout<<"Data type       Range "<<endl;
repchar('-', 20);
cout<<"char      -128 to 127"<<endl;
cout<<"short     -32,768 to 32, 767"<<endl;
cout<<"int       System dependent"<<endl;
cout<<"long      -2,147,483,648 to 2,147,483,647"<<endl;
repchar('-', 40);
cout<<"I Love And Every Body Loves C++"<<endl;
repchar('=', 50);
getch();
}
void repchar(char ch, int n)
{
for(int j=0; j<=n; j++)
cout<<ch;
cout<<endl;
}
```

هن پروگرام جو آئوٽ پٽ ڪجهه هن ريت نڪرندو.



پروگرام نمبر 47 جي آئوٽ پُٽ ڏانهن ڏسو، جنهن ۾ متي 50 = پرنٽ ٿيل آهن، ان کان پوءِ وري 20 ۽ پوءِ 40 ڊيش پرنٽ ٿيل آهن، ۽ آخر ۾ وري 50 برابر جون نشانيون پرنٽ ٿيل آهن، متي اڪثر سيني جا ڪريڪٽر ۽ ويليوز چينئج آهن.

پروگرام نمبر 47 ۾ هڪ فنڪشن repchar(); استعمال ٿيل آهي، جنهن ۾ هڪ انٽيجر ۽ هڪ ڪريڪٽر ويري ايبل استعمال ٿيل آهن. ڪريڪٽر ويري ايبل ۾ اسان (=,-) رکيا آهن ۽ انٽيجر ويري ايبل ۾ اسان مختلف ويليوز جا (=,-) پرنٽ ڪرايل آهن.

فنڪشن ڊيفنيشن ۾ فار لوپ جي ٽيسٽ ايڪسپريشن ۾ ويليو n رکيل آهي جيڪو اسان مختلف وقت جي چينج ڪندا رهيا آهيون.

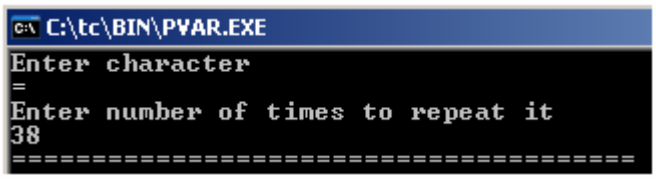
پاسنگ ويري ايبل Passing variabe:

اسان پروگرام نمبر 47 ۾ - ۽ 50 وغيره آگومنٽ طور رکيا آهن، هاڻي اسان پاسنگ ويري ايبل جو مثال ونڊاسين.

پروگرام نمبر 48

```
#include<constream.h>
void repchar(char,  int);
void main()
{
clrscr();
char chain;
int nin;
cout<<"Enter character"<<endl;
cin>>chain;
cout<<"Enter number of times to repeat it"<<endl;
cin>>nin;
repchar(chain, nin);
getch();
}
void repchar(char ch, int n)
{
for(int j=0; j<=n; j++)
cout<<ch;
cout<<endl;
}
```

هن پروگرام جو آئوٽ پٽ ڪجهه هن ريت نڪرندو.



هن پروگرام جي آئوٽ پُٽ کي سمجهڻ لاءِ پروگرام نمبر 47 کي پڙهو. هن پروگرام ۽ 47 نمبر ۾ صرف هڪ فرق آهي ته پروگرام نمبر 47 ۾ هڪ ڪريڪٽر مقرر هو پر هن ۾ يوزر سي اين جي ذريعي پاڻ ڪريڪٽر ۽ پرنٽگ نمبر ڪمپائيلر کي ٻڌائيندو.

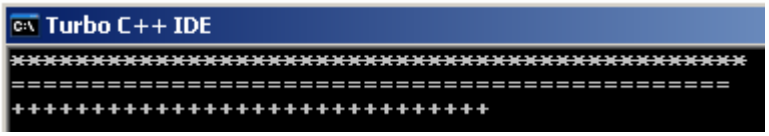
ڊيفرنٽ نمبر آف آرگومنٽDiff: no of argument؛

اسان اڳئي پروگرام نمبر 45، 46، 47، 48 ۾ هڪ فنڪشن ته ڪيڏهن هڪ آرگومنٽ استعمال ڪيوسين، پر هن پروگرام ۾ اسان انهن ٽنهي پروگرامن کي گڏي هڪ پروگرام جي صورت ۾ پيش ڪنداسين، هن پروگرام ۾ مختلف قسم جا آرگومنٽ استعمال ٿيل هوندا.

پروگرام نمبر 49

```
#include<constream.h>
void repchar();
void repchar(char);
void repchar(char, int);
void main()
{
clrscr();
repchar();
repchar('=');
repchar('+', 30);
getch();
}
void repchar()
{
for(int j=0; j<=45; j++)
cout<<'*';
cout<<endl;
}
void repchar(char ch)
{
for(int j=0; j<45; j++)
cout<<ch;
cout<<endl;
}
void repchar(char ch, int n)
{
for(int j=0; j<n; j++)
cout<<ch;
cout<<endl;
}
```

هن پروگرام جو آئوٽ پٽ ڪجهه هن ريت نڪرندو.



هنن پروگرامز جي بابت اوهان کي مٿي 45، 46، 47، 48 نمبر پروگرام ۾ هڪ هڪ ڪري سمجهايو ويو آهي، هتي صرف اوهان جي ڪانسيپٽ ڪليئر ڪرڻ لاءِ هي پروگرام ڏنو ويو آهي. مون کي يقين آهي ته اوهان مٿين پروگرامن کي ڏسي هن پروگرام کي باآساني سان سمجهي سگهو ٿا.

باب ڏهون
ورڪ وِٿ مُور
Work with more

هاڻي اچو تہ انهن فنڪشن ڏانهن هلون جن جي باري ۾ اسان مٿين پروگرامز ۾ سکي نہ سگهيا سين.

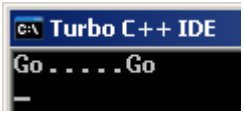
ٽيڪسٽ مُوڊ textmode :

هن فنڪشن جي ذريعي اوهان پروگرام ۾ اکرن جي سائيز وڌائي سگهو ٿا.
اچو تہ ٽيڪسٽ موڊ فنڪشن جو هڪ مثال ونون.

پروگرام نمبر 50

```
#include<constream.h>
int main()
{
clrscr();
textmode(3);
cout<<"Go.....Go"<<endl;
getch();
}
```

هن پروگرام جو آئوٽ پُٽ ڪجهه هن ريت نڪرندو.



نوٽ: هتي اکرن جي سائيز ۾ تبديلي ڏيکاريل ناهي پر سي پلس پلس ۾ اوهان هي فنڪشن استعمال ڪري اکرن ۾ تبديلي آڻي سگهو ٿا.

Textmode(3); هن فنڪشن ۾ جتي 3 لکيل نظر اچي رهيو آهي اتي اوهان پنهنجي مرضي مطابق اکرن جي سائيز رکي سگهو ٿا پر 3 استينڈرڊ سائيز آهي سي پلس پلس جي اندر.

ٽيڪسٽ ڪلر Text color :

ٽيڪسٽ ڪلر فنڪشن جي ذريعي اسان اسڪرين تي موجود اکرن جو رنگ تبديل ڪري سگهون ٿا. اچو تہ ٽيڪسٽ ڪلر فنڪشن جو هڪ مثال ونون.

پروگرام نمبر 51

```
#include<constream.h>
void main()
{
clrscr();
textcolor(12);
cputs("Information technology");
getch();
}
```

هن پروگرام جو آئوٽ پٽ ڪجهه هن ريت نڪرندو.



ٽيڪسٽ ڪلر ۾ ويليو 12 ڏنل آهي اوهان مختلف قسم جا ويليوز ڏئي اکرن جو رنگ تبديل ڪري سگهو ٿا.
ٽيڪسٽ ڪلر فنڪشن سان سي آئوٽ cout جي جڳهه تي سي پُٽس cputs استعمال ٿيندو ۽ انسرشن آپريٽر جي جڳهه تي بريسز استعمال ٿيندا.
اچو ته هاڻي معلوم ڪريون ته ڪهڙو ڪلر ڪهڙي انٽيجر ويليو ۾ آهي.

ويليو	ڪلر
0	
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	

ٽيڪسٽ بيڪگرائونڊ Text beckground:

ٽيڪسٽ بيڪگرائونڊ فنڪشن جي ذريعي اسان اڪرن جي هينيان هڪ بيو ڪلر ڏئي سگهون ٿا، اهي ڪلر مٽين ٽيبل مطابق انٽيجر ويليوز ذريعي سان ڏئي سگهجن ٿا.

پروگرام نمبر 52

```
#include<constream.h>
void main()
{
clrscr();
textcolor(12);
textbackground(2);
cputs("Information technology");
getch();
}
```

هن پروگرام جو آئوٽ پُٽ ڪجهه هن ريت نڪرندو.



هاءِ ويڊيو High video:

هاءِ ويڊيو فنڪشن جي مدد سان اسان سي پلس پلس ۾ اڪرن کي وڌيڪ شائينگ ڏئي سگهون ٿا.

پروگرام نمبر 53

```
#include<constream.h>
int main()
{
clrscr();
textmode(3);
highvideo();
cputs("Hello tech in sindhi!");
getch();
return 0;
}
```

هن پروگرام جو آئوٽ پُٽ ڪجهه هن ريت نڪرندو.



توهان ڪنهن آئوٽ پُٽ سان هن آئوٽ پُٽ کي ڪمپيئر ڪندا ته هنن اڪرن ۾ وڌيڪ شائينگ نظر ايندي.

نارمل ويڊيو Normal video:

نارمل ويڊيو فنڪشن سان اڪرن کي گهٽ شائينگ ڏئي سگهون ٿا.

پروگرام نمبر 54

```
#include<constream.h>
int main()
{
clrscr();
textmode(3);
normvideo();
cputs("Hello tech in sindhi!");
getch();
return 0;
}
```

لو وڊيو Low video:

لو وڊيو فنڪشن جي مدد سان اوهان اڪرن ڪان مڪمل طرح شائينگ ختم ڪري سگهو ٿا.

پروگرام نمبر 55

```
#include<constream.h>
int main()
{
clrscr();
textmode(3);
lowvideo();
cputs("Hello tech in sindhi!");
getch();
return 0;
}
```

clreol(); فنڪشن:

Clreol شارٽ فارم آهي clear end of line جو. هن جي مدد سان اوهان لائين جي آخر وارا اکر ڪٽي سگهو ٿا.

پروگرام نمبر 56

```
#include <constream.h>
void main()
{
   clrscr();
   cputs("clreol stand fo clear end of line\r\n");
   cputs("c plus plus\r\n");
   cputs("current text window, without moving the cursor.\r\n");
   cputs("Press any key to continue . . .");
   gotoxy(14, 4);
   getch();

   clreol();
   getch();
}
```

گو ٽو ايڪس واءِ goto xy ؛

گوٽو ايڪس واءِ فنڪشن جي ذريعي اوهان اسڪرين جي ڪنهن بہ حصي تہ استيٽمنٽ پرنٽ ڪرائي سگهو ٿا.

پروگرام نمبر 57

```
#include <constream.h>
void main()
```

```
{
clrscr();
gotoxy(12, 12);
cout<<"Hello world";
getch();
}
```

Delline(); فنڪشن:

Delline شارٽ فارم آهي delete line جو. هن جي ذريعي اوهان هڪ لائين ڪٽي سگهو ٿا. گوٽو ايڪس واءِ جي مدد سان اوهان ڪرسر کي جتي به سيٽ ڪندا اها لائين اوهان جي انٽر پريس ڪرڻ سان ئي ختم ٿي ويندو.

پروگرام نمبر 58

```
#include <constream.h>
void main()
{
clrscr();
cout<<"Information technology"<<endl;
cout<<"C plus pluse"<<endl;
cout<<"Hello world"<<endl;
cout<<"window.  Press any key to continue . . ."<<endl;
gotoxy(1,2);
getch();
delline();
getch();
}
```

Insline(); فنڪشن:

Insline شارٽ فارم آهي insert line جو شارٽ فارم آهي اوهان هن جي مدد سان پنهنجي پروگرام ۾ هڪ نئون لائين ايڊ ڪري سگهو ٿا.

پروگرام نمبر 60

```
#include <constream.h>
void main()
{
clrscr();
cout<<"INSLINE inserts an empty line in the text window"<<endl;
cout<<"at the cursor position using the current text"<<endl;
cout<<"background color.  All lines below the empty one"<<endl;
cout<<"move down one line and the bottom line scrolls"<<endl;
cout<<"off the bottom of the window."<<endl;
cout<<"Press any key to continue:";
gotoxy(1, 3);
getch();
insline();
getch();
}
```

ونڈوز فنڪشن windows function :

هي پڻ گوٽو ايڪس واءِ goto xy وانگر ڪم ڪندو آهي.

پروگرام نمبر 61

```
#include <conio.h>
#include <iostream.h>
int main(void)
{
   clrscr();
   window(10,10,40,91);
   textcolor(BLACK);
   textbackground(WHITE);
   cputs("This is a test\r\n");
   getch();
   return 0;
}
```

هن پروگرام ۾ ونڊوز جي اندر آرگومنٽ ڏنل آهن اهي اوهان پنهنجي مرضي مطابق رکي سگهو ٿا.

دلي، سائونڊ، نو سائونڊ Delay,sound no sound :

اچو ٽن فنڪشن جو گڏي مثال ونون.

پروگرام نمبر 62

```
#include <dos.h>
int main(void)
{
   sound(440);
   delay(1000);
   nosound();
   return 0;
}
```

هن پروگرام ۾ هڪ نئون هيڈر فائيل dos.h استعمال ڪيو ويو آهي. جيڪو هنن ٽنهن فنڪشن کي سپورٽ ڪري ٿو.

سائونڊ جي ذريعي اوهان مختلف قسم جا آواز وڻي سگهو ٿا. 440 فريڪنسي کي چيئنج ڪري مختلف فريڪنسي ڏئي اوهان اهو ڪم ڪري سگهو ٿا.

دلي فنڪشن جي مدد سان اوهان هڪ پروگرام کي پنهنجي مرضي آهر اسڪرين تي بيهاري سگهو ٿا. مثال مٿي 1000 ملي سيڪنڊ لکيل آهي، اوهان اهو گهٽ يا وڌ ڪري سگهو ٿا.

نو سائونڊ فنڪشن مٿي ڏنل سائونڊ کي پروگرام جي بند ٿيڻ شرط ئي ختم ڪري ڇڏيندو.

اسڪوائر روٽ Square root :

اسڪوائر روٽ فنڪشن جي ذريعي اوهان ڪنهن به انگ جو اسڪوائر روٽ معلوم ڪري سگهو ٿا.

پروگرام نمبر 63

```
#include<constream.h>
#include<math.h>
void main()
{
clrscr();
int no, result;
cout<<"Enter any number for square root"<<endl;
cin>>no;
result=sqrt(no);
cout<<"Square root is"<<result;
getch();
}
```

اسڪوائر روٽ لاءِ الڳ هيڈر فائيل math.h استعمال ٿيندو آهي.

پاور فنڪشن power function :

هن جي ذريعي اوهان ڪو به انگ ۽ ان جي مٿان ڏنل پاور سي پلس پلس ۾ ڏئي ان جو رزلٽ معلوم ڪري سگهو ٿا.

پروگرام نمبر 64

```
#include<constream.h>
#include<math.h>
void main()
{
clrscr();
int no, result,power;
cout<<"Enter any value"<<endl;
cin>>no;
cout<<"Enter any power"<<endl;
cin>>power;
result=pow(no,power);
cout<<"power is"<<result;
getch();
}
```

پاور فنڪشن به math.h هيڈر فائيل ۾ آهي.

هن فنڪشن ۾ اوهان کي power فنڪشن pow ڪري لکڻو آهي.

باب يارهون
آسڪي چارٽ

ASCII Chart

DEC	HEX	Symbol	Key	Use in c
0	00	(NULL)	Ctrl 2	
1	01	☺	Ctrl A	
2	02	☻	Ctrl B	
3	03	♥	Ctrl C	
4	04	♦	Ctrl D	
5	05	♣	Ctrl E	
6	06	♦	Ctrl E	
7	07	•	Ctrl G	beep
8	08	◘	Backspace	Backspace
9	09	○	Tab	tab
10	0A	◙	Ctrl J	New line
11	0B	♂	Ctrl K	Vertical tab
12	0C	♀	Ctrl L	Form feed
13	0D	♪	Enter	**Carriage return**
14	0E	♫	Ctrl N	
15	0F	☼	Ctrl O	
16	10	►	Ctrl P	
17	11	◄	Ctrl Q	
18	12	↕	Ctrl R	
19	13	‼	Ctrl S	
20	14	¶	Ctrl T	
21	15	§	Ctrl U	
22	17	↨	Ctrl W	
23	18	↑	Ctrl X	
24	19	↓	Ctrl Y	
25	1A	→	Ctrl Z	
26	1B	←	Esc	
27	1C	∟	Ctrl \	
28	1D	↔	Ctrl]	
29	1E	▲	Ctrl 6	
30	1F	▼	Ctrl -	
31	20		Spacebar	
32	21	!	!	

33	22	"	″	
34	23	#	#	
35	24	$	$	
36	25	%	%	
37	26	&	&	
38	27	'	′	
39	28	)	)	
40	27	(	(	
41	2A	*	*	
42	2B	+	+	
43	2C	,	,	
44	2D	-	-	
45	2E	.	.	
46	2F	/	/	
47	30	0	0	
48	31	1	1	
49	32	2	2	
50	33	3	3	
51	34	4	4	
52	35	5	5	
53	36	6	6	
54	37	7	7	
55	38	8	8	
56	39	9	9	
57	3A	:	:	
58	3B	;	;	
59	3C	>	>	
60	3D	=	=	
61	3E	<	<	
62	3F	?	?	
63	40	@	@	
64	41	A	A	
65	42	B	B	
66	43	C	C	
67	44	D	D	
68	45	E	E	
69	46	F	F	
70	47	G	G	
71	48	H	H	
72	49	I	I	

73	4A	J	J	
74	4B	K	K	
75	4C	L	L	
76	4D	M	M	
77	4E	N	N	
78	4F	O	O	
79	50	P	P	
80	51	Q	Q	
81	52	R	R	
82	53	S	S	
83	54	T	T	
84	55	U	U	
85	56	V	V	
86	57	W	W	
87	58	X	X	
88	59	Y	Y	
89	5A	Z	Z	
90	5B	]	]	
91	5C	\	\	
92	5D	[	[	
93	5E	^	^	
94	5F	_	_	
95	60	`	`	
96	61	a	a	
97	62	b	b	
98	63	c	c	
99	64	d	d	
100	65	e	e	
101	66	f	f	
102	67	g	g	
103	68	h	h	
104	69	i	i	
105	6A	j	j	
106	6B	k	k	
107	6C	l	l	
108	6D	m	m	
109	6E	n	n	
110	6F	o	o	
111	70	p	p	
112	71	q	q	

113	72	r	r	
114	73	s	s	
115	74	t	t	
116	75	u	u	
117	76	v	v	
118	77	w	w	
119	78	x	x	
120	79	y	y	
121	7A	z	z	
122	7B	{	{	
123	7C	|	|	
124	7D	}	}	
125	7E	~	~	
126	7F	Δ	Ctrl ←	
127	80	Ä	Alt 128	
128	81	ü	Alt 129	
129	82	é	Alt 130	
130	83	É	Alt 131	
131	84	ä	Alt 132	
132	85	à	Alt 133	
133	86	å	Alt 134	
134	87	ç	Alt 135	
135	88	ê	Alt 136	
136	89	ë	Alt 137	
137	8A	è	Alt 138	
138	8B	ï	Alt139	
139	8C	î	Alt 140	
140	8D	ì	Alt 141	
141	8E	Ä	Alt 142	
142	8F	Å	Alt 143	
143	90	É	Alt 144	
144	91	æ	Alt 145	
145	92	Æ	Alt 146	
146	93	Ô	Alt 147	
147	94	Ö	Alt 148	
148	95	Ò	Alt 149	
149	96	û	Alt 150	
150	97	ù	Alt 151	
151	98	ÿ	Alt 152	
152	99	Ö	Alt 153	

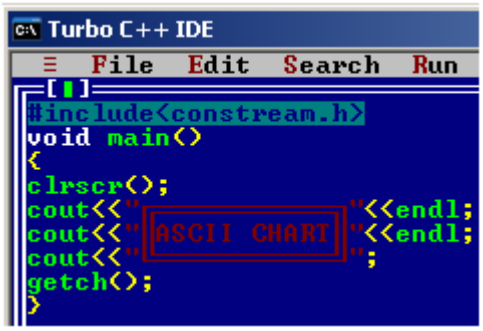
153	9A	Ü	Alt 154	
154	9B	õ	Alt 155	
155	9C	£	Alt 156	
156	9D	¥	Alt 157	
157	9E	û	Alt 158	
158	9F	ü	Alt 159	
159	A0	á	Alt 160	
160	A1	í	Alt 161	
161	A2	ó	Alt 162	
162	A3	ú	Alt 163	
163	A4	ñ	Alt 164	
164	A5	Ñ	Alt 165	
165	A6	ª	Alt 166	
166	A7	º	Alt 167	
167	A8	¿	Alt 168	
168	A9	⌐	Alt 169	
169	AA	¬	Alt 170	
170	AB	½	Alt 171	
171	AC	¼	Alt 172	
172	AD	¡	Alt 173	
173	AE	«	Alt 174	
174	AF	»	Alt 175	
175	B3	┃	Alt 179	
176	B4	┤	Alt 180	
177	B5	╡	Alt 181	
178	B6	╢	Alt 182	
179	B7	╖	Alt 183	
180	B8	╕	Alt 184	
181	B9	╣	Alt 185	
182	BA	║	Alt 186	
183	BB	╗	Alt 187	
184	BC	╝	Alt 188	
185	BD	╜	Alt 189	
186	BE	╛	Alt 190	
187	BF	┐	Alt 191	
188	C0	└	Alt 192	
189	C1	┴	Alt 193	
190	C2	┬	Alt 194	
191	C3	├	Alt 195	
192	C4	─	Alt 196	

193	C5	┼	Alt 197	
194	C6	╞	Alt 198	
195	C7	╟	Alt 199	
196	C8	╚	Alt 200	
197	C9	╔	Alt 201	
198	CA	╩	Alt 202	
199	CB	╦	Alt 203	
200	CC	╠	Alt 204	
201	CD	═	Alt 205	
202	CE	╬	Alt 206	
203	CF	╧	Alt 207	
204	D0	╨	Alt 208	
205	D1	╤	Alt 209	
206	D2	╥	Alt 210	
207	D3	╙	Alt 211	
208	D4	╘	Alt 212	
209	D5	╒	Alt 213	
210	D6	╓	Alt 214	
211	D7	╫	Alt 215	
212	D8	╪	Alt 216	
213	D9	┘	Alt 217	
214	DA	┌	Alt 218	
215	DB	█	Alt 219	
216	DC	▄	Alt 220	
217	DD	▌	Alt 221	
218	DE	▐	Alt 222	
219	DF	▀	Alt 223	
220	E0	α	Alt 224	
221	E1	β	Alt 225	
222	E2	Γ	Alt 226	
223	E3	π	Alt 227	
224	E4	Σ	Alt 228	
225	E5	σ	Alt 229	
226	E7	τ	Alt 231	
227	E8	ϕ	Alt 232	
228	E9	Θ	Alt 233	
229	EA	Ω	Alt 234	
230	EB	δ	Alt 235	
231	EC	∞	Alt 236	
232	ED	φ	Alt 237	

233	EE	∈	Alt 238	
234	EF	∩	Alt 239	
235	F0	≡	Alt 240	
236	F1	±	Alt 241	
237	F2	≥	Alt 242	
238	F3	≤	Alt 243	
239	F4	⌠	Alt 244	
240	F5	⌡	Alt 245	
241	F6	÷	Alt 246	
242	F7	≈	Alt 247	
243	F8	∘	Alt 247	
244	F9	•	Alt 249	
245	FA	·	Alt 250	
246	FB	√	Alt 251	
247	FC	η	Alt 252	
248	FD	²	Alt 253	
249	FE	■	Alt 254	
250	FF	(blank)	Alt 255	

ASCII شارٽ فارم آهي Amrican standard code for information interchange جو هن جي ذريعي اوهان مختلف نشانيون، روم ڪريڪٽر، ۽ ٻيون ڪوڙ ساريون شيون لکي سگهندا جيڪي عام ڪي بورڊ جي زريعي لکڻ ناممڪن آهن.

آسڪي جا پروگرام اوهان ڪجهه هن ريت لکي سگهو ٿا.



هن پروگرام جو آئوٽ پُٽ ڪجهه هن ريت نڪرندو.



جيئن مٿي بن اکرن کي پاسي کان باربر ڏنو ويو آهي. اوهان بہ اهڙا مختلف قسم جا گرافڪس ترتيب ڏئي سگهو ٿا.

اوهان کي سيمبل جا شارٽڪٽ کي ٻڌايل آهن، مثال فارمفيد جي لاءِ ctrl L يا وري سگما لاءِ alt 228 وغيره.

باب بارهون
آئوٽ پٽ سچاٽڻ

هن باب ۾ ڪجهه پروگرام ڏنا ويا آهن اوهان سي پلس پلس ۾ وڃڻ بنا هنن جي آئوٽ پُٽ کي سچاٽڻ جي ڪوشش ڪريو.

هيٺ جيڪي به پروگرام ڏنا ويا آهن انهن جي هر هڪ فنڪشن بابت اوهان کي تفصلي ڄاڻ ڏني وئي آهي، هاڻي اوهان جو ڪم آهي ته اسان جي بغير ڪنهن مداخلت جي اوهان آئوٽ پٽ ڳولي لهو.

پروگرام نمبر 66

```
#include<constream.h>
void main()
{
clrscr();
int choice;
cout<<"Choice one from following"<<endl;
cout<<"1. Fahrenheit to Centigrad"<<endl;
cout<<"2. Centigrad to Fahrenheit"<<endl;
cin>>choice;
switch(choice)
{
case 1:
float centi, forn;
cout<<"Enter temp: in fahrenheit"<<endl;
cin>>forn;
centi=(forn-32)*5/9;
cout<<"Centigrad is "<<centi;
break;

case 2:
float cnti, fahr;
cout<<"Enter temp: in centigrad"<<endl;
cin>>cnti;
fahr=cnti*9/5+32;
cout<<"Fahrenheit is "<<fahr;
break;
default:
cout<<"Your input not valid";
}
getch();
}
```

پروگرام نمبر67

```
#include<constream.h>
void main()
{
int notes;
clrscr();
cout<<"Enter value for notes"<<endl;
cin>>notes;
if(notes>=1000)
{
cout<<"Notes of one thousands are "<<notes/1000<<endl;
notes=notes%1000;
}
if(notes>=500)
```

```
{
cout<<"Notes of five hundred are "<<notes/500<<endl;
notes=notes%500;
}
if(notes>=100)
{
cout<<"Notes of one hundred are "<<notes/100<<endl;
notes=notes%100;
}
if(notes>=50)
{
cout<<"Notes of fifty are"<<notes/50<<endl;
notes=notes%50;
}
if(notes>=10)
{
cout<<"Notes of ten are"<<notes/10<<endl;
notes=notes%10;
}
if(notes>=5)
{
cout<<"Notes of five are "<<notes/5<<endl;
notes=notes%5;
}
if(notes>=0)
{
cout<<"notes of one are "<<notes/1<<endl;
notes=notes%1;
}
getch();
}
```

پروگرام نمبر 68

```
#include<constream.h>
void main()
{
clrscr();
int choice;
cout<<"Choice one from following"<<endl;
cout<<"1. Inchs To Feets"<<endl;
cout<<"2. Feets to inchs"<<endl;
cin>>choice;
switch(choice)
{
case 1:
float inchs, feet;
cout<<"Enter Feets"<<endl;
cin>>feet;
inchs=feet*12;
cout<<"Inch(s) are "<<inchs;
break;

case 2:
float inch, feets;
cout<<"Enter Inchs"<<endl;
cin>>inch;
feet=inch/12;
cout<<"Feet(s) are"<<feets;
break;
default:
```

```
cout<<"Your input not valid";
}
getch();
}
```

پروگرام نمبر 69

```
#include<constream.h>
void main()
{
clrscr();
int table;
cout<<"Enter any table"<<endl;
cin>>table;
for(int a=1; a<=10; a++)
cout<<table<<"*"<<a<<"="<<a*table<<endl;
getch();
}
```

پروگرام نمبر 70

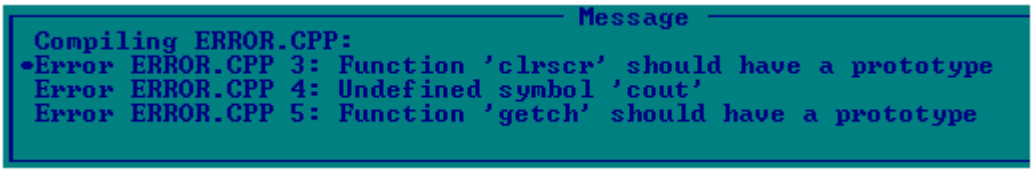
```
#include<constream.h>
void main()
{
int eng, sindhi, ps, science, drawing, math;
float per, total;
clrscr();
textmode(3);
cout<<"Enter marks of English"<<endl;
cin>>eng;
cout<<"Enter marks of Sindhi"<<endl;
cin>>sindhi;
cout<<"Enter marks of Pakistan study"<<endl;
cin>>ps;
cout<<"Enter marks of Science"<<endl;
cin>>science;
cout<<"Enter marks of Drawing"<<endl;
cin>>drawing;
cout<<"Enter marks of Math"<<endl;
cin>>math;
total=eng+sindhi+ps+science+drawing+math;
per=total*100/600;
cout<<"Total marks are 600"<<endl;
cout<<"Your marks are "<<total<<endl;
cout<<"Your percantage is "<<per<<endl;
if(per>=80 && per<=100)
cout<<"Your grad is A"<<endl;
else if(per>=60 && per<=80)
cout<<"Yoyr grad is B"<<endl;
else if(per>=50 && per<=60)
cout<<"Your grad is C"<<endl;
else if(per>=40 && per<=50)
cout<<"Your grad is D"<<endl;
else if(per<40)
cout<<"Your are fail"<<endl;
getch();
}
```

باب تيرهون
ايرر رپورٽنگ
Error reporting

عام زندگي وانگر انسان کان پروگرامنگ مهل ۾ غلطيون ٿيندیون آهن، عام زندگي جي غلطي ناقابلي تلافي ۾ هوندي آهي پر پروگرامنگ ۾ سوين غلطيون ڪري ان کي منٽ اندر صحيح ڪري سگهجي ٿو.

ايرر رپوٽنگ1:

اڪثر ڪري اوهان جڏهن پروگرام ۾ هيڈر فائيل استعمال نه ڪندا آهيو ته پوءِ هيٺيان قسم جا ايرر اڪثر ڪري اوهان کي ڊسٽرب ڪندا آهن.



مٿي 3،4،5 لکيل آهن اِهي پروگرام ۾ اُهي ست آهن جي غلطي جي ڪري ايگزيڪوٽ نه ٿي سگهيا آهن.

Function clrscr should have prototype هن ايرر ۾ ٻڌايل آهي ته جيڪو اوهان clrscr نالي فنڪشن ڏنو آهي ان جي ڊيفينيشن ڏنل ناهي. جڏهن اسان هيڈر فائيل ڏيندا آهيون ته انهي ۾ هنن فنڪشنن جو ديفينيشن اڳئي سيو ٿيل هوندو آهي.

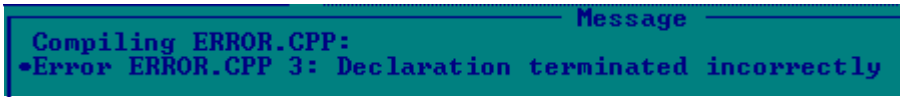
Undefined symbol cout هن جو مطلب آهي ته ڪمپائيلر سي آئوٽ کي هڪ ويري ايبل طور ڪنسيڊر ڪيو وينو آهي، چو ته هن جي سجاڻ ڪرڻ وارو هيڈر فائيل موجود ناهي.

Getch() سان پڻ ساڳيو حشر آهي.

ايرر رپوٽنگ 2:

اڪثر ڪري سيمي ڪالن يا وري ڪاما جو غلط استعمال ڪرڻ سان هن قسم جا ايرر ايندا آهن.

مثال: int main();



ايرر رپوٽنگ 3:

هن قسم جا ايرر اڪثر ڪري پروگرام جي لکڻ جي طريقي ۾ غلطي سبب ايندا آهن.

مثال: it main(); ;

```
Message
Compiling ERROR.CPP:
•Error ERROR.CPP 2: Declaration syntax error
```

ايرر رپوٽنگ 4:

هن قسم جا ايرر ان وقت ايندا آهن جڏهن اسان بريسز جو غلط استعمال ڪيون.

```
Error ERROR.CPP 8: Unexpected }
```

ايرر رپوٽنگ 5:

هن قسم جا ايرر اڪثر ڪري ان وقت ايندا آهن جڏهن اسان لائين ٽرمينٽر جو استعمال نہ ڪيون.

مثال: clrscr() ;

```
Message
Compiling ERROR.CPP:
•Error ERROR.CPP 5: Statement missing ;
```

ايرر رپوٽنگ 6:

هن قسم جا ايرر ان وقت ايندا آهن جڏهن اسان اسٽيٽمنٽ ۾ ڪا هڪ ڪاما يا ٻئي ڪوڊ کي ڇڏي ڏيو.

مثال: cout<”Hello”; ;

```
Message
Compiling ERROR.CPP:
•Warning ERROR.CPP 5: Code has no effect
Linking ERROR.EXE:
```

ايرر رپوٽنگ7:

هن قسم جا ايرر ان وقت ايندا آهن جڏهن اسان اسٽرنگ يا ڪريڪٽر ڪانسٽنٽ جي لاءِ سنگل ٽوڙي ڊبل ڪوٽيشن جو غلط استمال ڪيون.

مثال: cout<<hello”; ;

```
Error ERROR.CPP 5: Unterminated string or character constant
```

هي چند اهڙا عام ايرر آهن جيڪي اڪثر ڪري هر پروگرام ۾ ايندا آهن. انهي لاءِ هنن عام ايرر جي بابت اوهان کي ٻڌائڻ پنهنجو فرض سمجهيم پر عام طور تہ پروگرام کي هڪ چڪر غور سان ڏسڻ شرط سي ايرر نظر اچي ويندا آهن.

پوروٽيو

www.voiceofsindh.net