یونٹ نمبر 2
یوزر انٹریکشن

س: کمپیوٹر اور آؤٹ پٹ فنکشن کیا ہے؟

کمپیوٹر ایک ایسا آلہ ہے جو ڈیٹا کو ان پٹ کے طور پر لیتا ہے ڈیٹا کو پروسیس کرتا ہے اور آؤٹ پٹ دیتا ہے۔ جبکہ آؤٹ پٹ فنکشن پروگرام سے ڈیٹا حاصل کرنے کے لیے استعمال ہوتا ہے۔ مانیٹر کے ذریعہ تیار شدہ آؤٹ پٹ کو معیاری آؤٹ پٹ کہا جاتا ہے۔ سی لینگویج آؤٹ پٹ ظاہر کرنے کے لیے printf فنکشن مہیا کرتی ہے۔

س: سی لینگویج کے () printf فنکشن کو ایک مثال کے ساتھ بیان کریں۔ (یا) سکرین پر آؤٹ پٹ ظاہر کرنے کے لیے سی لینگویج کا کون سا فنکشن استعمال کیا جاتا ہے؟

printf ()

printf سکرین پر آؤٹ پٹ دکھانے کے لیے سی پروگرامنگ لینگویج میں ایک بلٹ ان (built-in) فنکشن ہے۔ اس کا نام "پرنٹ فارمیٹڈ" (print formatted) سے نکلا ہے جو فارمیٹ شدہ آؤٹ پٹ کو سکرین پر پرنٹ کرنے کے لیے استعمال ہوتا ہے۔ تمام ڈیٹا کی اقسام printf فنکشن کے ساتھ ظاہر کی جاسکتی ہیں۔ printf فنکشن کے کام کو سمجھنے کے لیے درج ذیل پروگرامنگ مثال پر غور کریں:

مثال:

Program	Output:

#include<stdio.h>
void main()
{
printf("Hello World");
}
	Hello World

taleemcity.com

اس مثال میں printf فنکشن Hello World کو سکرین پر دکھانے کے لیے استعمال ہوتا ہے۔ ہم () printf فنکشن میں واوین (" ") کے اندر جو کچھ بھی لکھتے ہیں سکرین پر ظاہر ہوتا ہے۔

سوال: I / O (ان پٹ / آؤٹ پٹ) آپریشنز میں فارمیٹ سپیسفائر کی وضاحت کیوں اہم ہیں؟

فارمیٹ سپیسفائر ڈیٹا ٹائپ فیلڈ کی چوڑائی اور سکرین پر ظاہر ہونے والے متغیر کی ویلیو کی شکل کی نمائندگی کرتا ہے۔ ایک فارمیٹ سپیسفائر ہمیشہ علامت % سے شروع ہوتا ہے۔ فارمیٹ سپیسفائر ان پٹ اور آؤٹ پٹ دونوں سٹیٹمنٹ کے لیے استعمال ہوتا ہے۔ فارمیٹ سپیسفائر کا عمومی سنٹیکس ہے۔ %

مثال: اگر ہم کسی متغیر کی قیمت کو ظاہر کرنا چاہتے ہیں؟ ایک متغیر کو ڈیکلیئر کریں اور پھر printf کے رویے کو چیک کریں۔

```
int age = 35;
printf ("age");
```

اب اگر ہم اس متغیر age کی قیمت سکرین پر ظاہر کرنا چاہتے ہیں۔ تو مندرجہ ذیل سٹیٹمنٹ لکھیں گے:

FSC, Matric Notes, Test Papers, Past Papers & other study matrial, available on taleemcity.com

لیکن یہ مقصد کو پورا نہیں کرتا کیونکہ یہ سکرین پر درج ذیل آؤٹ پٹ دکھاتا ہے۔

age

یہ متغیر age کے اندر محفوظ شدہ قیمت کو ظاہر نہیں کرتا ہے اس کے بجائے یہ صرف وہی دکھاتا ہے جو printf کے واوین کے اندر لکھا گیا تھا۔

درحقیقت ہمیں فارمیٹ سپیسفائر کی وضاحت کرنے کی ضرورت ہے جسے ہم سکرین پر دکھانا چاہتے ہیں۔ مندرجہ ذیل ٹیبل سی لینگویج میں مختلف ڈیٹا ٹائپس کے مقابلے میں فارمیٹ سپیفائر دکھاتا ہے۔

Data type	Format Specifier
int	% d or % i
float	% f
char	% c

فرض کریں کہ ہم int ٹائپ کا ڈیٹا دکھانا چاہتے ہیں ہمیں اسے printf کے اندر واضع کرنے کے لیے d % یا i % استعمال کرنا چاہیے۔ اسی طرح float ٹائپ ڈیٹا کے لیے ہمیں f % کا استعمال کرنا چاہیے۔ یہ مندرجہ ذیل مثال میں دکھایا گیا ہے:

Program

```
#include<stdio.h>
#include<conio.h>
void main()
{
    clrscr();
    float height = 5.8;
    int age = 35;
    printf(" My age is %d and my height is %f ", age, height);
    getch();
}
```

Output:

My age is 35 and my height is 5.800000

taleemcity.com

ہم دیکھ سکتے ہیں کہ آؤٹ پٹ دکھاتے وقت پہلے فارمیٹ سپیسفائر کی جگہ واوین کے بعد دیے گئے ڈیٹا میں سے پہلے متغیر یعنی age کی قیمت آگئی ہے اور دوسرے فارمیٹ سپیسفائر کی جگہ دوسرا متغیر آ گیا ہے۔

جب ہم فلوٹ ویلیو ظاہر کرنے کے لیے f % کا استعمال کرتے ہیں تو یہ اعشاریہ کے بعد 6 ہندسے دکھاتا ہے۔ اگر ہم اعشاریہ کے بعد ہندسوں کی تعداد بتانا چاہتے ہیں تو ہم nf % لکھ سکتے ہیں جہاں n ہندسوں کی تعداد ہے۔ مندرجہ بالا مثال میں اگر ہم یہ سٹیٹمنٹ لکھیں۔

Program

```
#include<stdio.h>
#include<conio.h>
void main()
{
    clrscr();
    float height = 5.8;
    int age = 35;
    printf(" My age is %d and my height is %. 2f ", age, height);
    getch();
}
```

Output:

My age is 35 and my height is 5.80

فارمیٹ سپیسفائر صرف متغیر کے لیے استعمال نہیں ہوتے۔ دراصل ان کا استعمال متغیرات، کانسٹنٹ یا دونوں پر مشتمل کسی بھی ایکسپریشن کا نتیجہ ظاہر کرنے کے لیے کیا جاتا ہے جیسا کہ نیچے دی گئی مثال میں دکھایا گیا ہے۔

Program

```
#include<stdio.h>
#include<conio.h>
void main()
{
    clrscr();
    printf("Sum of 23 and 45 is %d", 23 + 45);
    getch();
}
```

Output

```
Sum of 23 and 45 is 68
```

س: scanf() پر نوٹ لکھیں؟

scanf() : scanf سی لینگویج میں ایک بلٹ ان فنکشن ہے جو صارف سے ویری ایبلز میں ان پٹ لیتا ہے۔ scanf فنکشن میں ہم فارمیٹ سپیسفائر کی مدد سے ان پٹ ڈیٹا کی متوقع قسم کی وضاحت کرتے ہیں۔ اگر صارف انٹیجر ڈیٹا ٹائپ میں ڈیٹا دیتا ہے تو scanf میں بیان کردہ فارمیٹ سپیسفائر d% یا i% ہونا چاہیے۔ مندرجہ ذیل مثال پر غور کریں۔

```
#include<stdio.h>
#include<conio.h>
void main()
{
        clrscr();
        char grade;
        scanf("%c", &grade);
        getch();
}
```

اس مثال میں c% فارمیٹ سپیسفائر کریکٹر ٹائپ ویری ایبل کے لیے استعمال کیا گیا ہے۔ صارف کی دی گئی ان پٹ کو متغیر **grade** میں محفوظ کیا گیا ہے۔ scanf فنکشن کے دو اہم حصے ہیں جس کو اوپر کوڈ میں دیکھا جا سکتا ہے۔ واوین کے اندر پہلا حصہ فارمیٹ سپیسفائر کی فہرست ہے اور دوسرا حصہ متغیرات کی فہرست ہے اور ان کے بائیں طرف & لکھا جاتا ہے۔

Program

```
#include<stdio.h>
#include<conio.h>
void main()
{
    clrscr();
    int number;
    printf("Enter a number between 0-10:");
    scanf("%d", &number);
    printf("The number you entered is: %d", number);
    getch();
}
```

Output

```
Enter a number between 0-10: 4
The number you entered is    : 4
```

taleemcity.com

ہم ایک ہی scanf فنکشن کا استعمال کرتے ہوئے متعدد ان پٹ لے سکتے ہیں۔ مندرجہ ذیل بیان پر غور کریں۔

scanf ("%d%d%f", &a, &b, &c);

یہ دو انٹیجر ویری ایبل a اور b اور ایک فلوٹ ویری ایبل c میں ان پٹ لیتا ہے۔ ہر ان پٹ کے بعد صارف کو ایک space یا enter دبانا ہوتا ہے۔ تمام ان پٹ کے بعد صارف کو enter دبانا لازمی ہوتا ہے۔ scanf فنکشن میں & اوپریٹر کو بھول جانا بہت عام سی غلطی ہے۔ & اوپریٹر کے بغیر پروگرام ایگزیکیوٹ تو جاتا ہے مگر متوقع نتائج نہیں دیتا ہے۔

سوال: ()getch پر نوٹ لکھیں؟

getch ()

() getch فنکشن صارف سے ایک کریکٹر لینے کے لیے استعمال کیا جاتا ہے۔ صارف کا داخل کردہ کریکٹر سکرین پر ظاہر نہیں ہوتا ہے۔ یہ فنکشن عام طور پر پروگرام کی ایگزیکیوشن کو روکنے کے لیے استعمال کیا جاتا ہے کیونکہ پروگرام اس وقت تک آگے نہیں چلتا جب تک کہ صارف کوئی بٹن نہ دبائے۔ اس فنکشن کو استعمال کرنے کے لیے ہمیں لائبریری conio.h کو پروگرام کے ہیڈر سیکشن میں شامل کرنے کی ضرورت پڑتی ہے۔

```
#include<stdio.h>
#include<conio.h>
void main()
{
        clrscr();
        printf("Enter any key if you want to exit program");
        getch();
}
```

درج بالا پروگرام صارف سے کوئی بھی کریکٹر ان پٹ لیتا ہے اور پھر پروگرام کے عمل کو مکمل کرنے سے پہلے صارف کے ان پٹ کا انتظار کرتا ہے۔

taleemcity.com

```
#include<stdio.h>
#include<conio.h>
void main()
{
        clrscr();
        char key;
        printf("Enter any key:");
        key = getch(); // Gets a character from user into variable key
        getch();
}
```

اگر ہم اس پروگرام کو چلاتے ہیں تو ہمیں scanf کا استعمال کرتے ہوئے ایک کریکٹر کو پڑھنے اور getch فنکشن کا استعمال کرتے ہوئے ایک کریکٹر کو پڑھنے میں فرق نظر آتا ہے۔ جب ہم scanf کے ذریعے کریکٹر پڑھتے ہیں تو ہمیں پروگرام کو آگے چلانے کے لیے enter دبانے کی ضرورت ہوتی ہے۔ لیکن getch کی صورت میں یہ enter کی کو دبانے کا انتظار نہیں کرتا ہے۔ فنکشن ایک کریکٹر کو پڑھتا ہے اور اگلی لائن کو چلانے کی طرف بڑھتا ہے۔

س: () clrscr پر نوٹ لکھیں؟

clrscr ()

یہ فنکشن سی لینگوئج ایڈیٹر کی آؤٹ پٹ سکرین کو صاف کرنے کے لیے استعمال ہوتا ہے۔

مثال:

Program

```
#include<stdio.h>
#include<conio.h>
void main()
{
    clrscr();
    int number;
    printf("Enter a number between 0-10:");
    scanf("%d", &number);
    printf("The number you entered is: %d", number);
    getch();
}
```

Output

```
Enter a number between 0-10: 4
The number you entered is    : 4
```

س: سٹیٹمنٹ ٹرمینیٹر کیا ہے؟

سٹیٹمنٹ ٹرمینیٹر (Statement Terminator)

سٹیٹمنٹ ٹرمینیٹر کمپائلر کے لیے شناخت کنندہ ہے جو لائن کے اختتام کی نشاندہی کرتا ہے۔ سی لینگوئج میں سیمی کولن (;) بطور سٹیٹمنٹ ٹرمینیٹر استعمال ہوتا ہے۔ اگر ہم ہر سٹیٹمنٹ کے آخر میں سیمی کولن نہ لگائیں تو ایرر (Error) آجاتا ہے۔

taleemcity.com

printf("Hello World") (;) (;) → Statement Terminattor

س: اسکیپ سیکوئنس کیا ہیں؟ ہمیں ان کی ضرورت کیوں ہے؟

اسکیپ سیکوئنس (Escape Sequence)

اسکیپ سیکوئنس printf فنکشن میں واوین ("") کے اندر استعمال ہوتے ہیں۔ جو printf کو استعمال کرتے ہوئے آؤٹ پٹ دکھانے کے اپنے طریقہ کار کو تبدیل کرنے پر مجبور کرتے ہیں۔ مثال:

```
printf ("My name is \"Ali\"");
```

مندرجہ بالا سٹیٹمنٹ کی آؤٹ پٹ یہ ہے۔

My name is "Ali"

مندرجہ بالا مثال میں " \ " ایک اسکیپ سیکوئنس ہے۔ یہ printf کو " کمپیوٹر اسکرین پر " ظاہر کرنے کا سبب بنتا ہے۔

اسکیپ سیکوئنس کی بناوٹ

اسکیپ سیکوئنس دو حروف پر مشتمل ہے۔ پہلا کریکٹر ہمیشہ بیک سلیش (back slash) ("\ ") ہوتا ہے اور دوسرا کریکٹر مطلوبہ فنکشنیلٹی کے مطابق آتا ہے جسے ہم حاصل کرنا چاہتے ہیں۔ بیک سلیش (\) کو اسکیپ کریکٹر کہا جاتا ہے جو اسکیپ کے بارے میں نشاندہی کرنے کے لیے ہر اسکیپ سیکوئنس سے منسلک ہوتا ہے۔ اسکیپ کریکٹر اور اس کے بعد آنے والا کریکٹر سکرین پر ظاہر نہیں ہوتے ہیں لیکن یہ مخصوص کام کرتے ہیں جو ان سے منسوب کیا گیا ہے۔ درج ذیل اسکیپ سیکوئنس عام طور پر C لینگوئج میں استعمال ہوتے ہے۔

سیکوئنس	مقصد	سیکوئنس	مقصد
\'	سنگل کوٹ سکرین پر دکھانا (')	\a	الرٹ کی آواز پیدا کرتا ہے
\\	بیک سلیش سکرین پر دکھانا (\)	\b	پچھلا کریکٹر مٹاتا ہے
\t	8 سپیس سکرین پر دکھانا	\n	اگلی لائن پر لے کر جاتا ہے

نیولائن (\n) (New line)

اسکیپ کریکٹر کے بعد n واضح کرتا ہے کہ کرسر (cursor) کو اگلی لائن (New Line) کے شروع پر لے کر جاتا ہے۔ یہ اسکیپ سیکوئنس ایک سے زیادہ لائنوں پر آؤٹ پٹ پرنٹ کرنے کے لیے استعمال ہوتا ہے۔ **مثال:**

Program

```
#include<stdio.h>
#include<conio.h>
void main()
{
      clrscr();
      printf("My name is Ali. \n");
      printf("I live in Lahore");
      getch();
}
```

Output

```
My name is Ali.
I live in Lahore
```

اسکیپ سیکوئنس کے بغیر اگر ہمارے پاس ایک سے زیادہ printf سٹیٹمنٹ بھی ہوں تو ان کی آؤٹ پٹ ایک ہی لائن پر ظاہر ہوگی۔ مندرجہ ذیل مثال یہ بات واضح کرتی ہے۔ **مثال:**

Program

```
#include<stdio.h>
#include<conio.h>
void main()
{
      clrscr();
      printf("My name is");
      printf("Ahmad");
      getch();
}
```

Output

```
My name is Ahmad
```

taleemcity.com

ٹیب "\t"

اسکیپ سیکوئنس t \ , I/O فنکشن کو بتاتا ہے کہ افقی طور پر اگلے ٹیب سٹاپ (Tab Stop) پر جانا ہے۔ ایک ٹیب اسٹاپ 8 سپیسز (spaces) کا مجموعہ ہوتا ہے۔ t\ کا استعمال کر سر کو اگلے ٹیب اسٹاپ پر لے جاتا ہے۔ **مثال:**

Program

```
#include<stdio.h>
#include<conio.h>
void main()
{
        clrscr();
        printf("Name: \tAli\nFname: \tHammad\nMarks: \t1000");
        getch();
}
```

Output

```
Name:  Ali
Fname: Hammad
Marks:  1000
```

س: آپریٹرز کیا ہیں؟

آپریٹرز (Operators)

آپریٹرز ریاضی کی علامتیں ہیں جو اوپرینڈ (operand) پر کچھ آپریشن کرتی ہیں۔ اوپرینڈز ویری ایبلز یا ویلیوز ہو سکتی ہیں۔ C میں بہت سے آپریٹرز استعمال ہوتے ہیں ان میں سے کچھ یہ ہیں:

- **اسائنمنٹ آپریٹر** (Assignment operator)
- **ارتھمیٹک آپریٹرز** (Arithmetic operators)
- **منطقی آپریٹرز** (Logical operators)
- **ریلیشنل آپریٹرز** (Relational operators)

س: اسائنمنٹ آپریٹر کی وضاحت کریں؟

اسائنمنٹ آپریٹر (Assignment Operator)

اسائنمنٹ آپریٹر ایک متغیر کو قیمت تفویض کرنے کے لیے استعمال کیا جاتا ہے یا ایک متغیر کی قیمت کو دوسرے متغیر میں رکھنے کے لیے استعمال ہوتا ہے۔ برابری کی علامت (=) کو سی لینگویج میں اسائنمنٹ آپریٹر کے طور پر استعمال کیا جاتا ہے۔ مثال:

```
int sum = 5;
```

کوڈ کی یہ لائن چلنے کے بعد ویری ایبل sum میں قیمت 5 محفوظ ہو جائے گی۔

```
int sum = 6;
int var = sum;
```

پہلے ویری ایبل sum کی قیمت 6 رکھی گی پھر اگلی لائن میں sum کی قیمت ویری ایبل var سے منسوب کر دی گی۔

FSC, Matric Notes, Test Papers, Past Papers & other study matrial, available on taleemcity.com

مثال: ایک پروگرام لکھیں جو دو انٹیجر ویری ایبلز کی قیمتوں کا آپس میں تبادلہ (swap) کرے۔

```
#include<stdio.h>
#include<conio.h>
void main()
{
        clrscr();
        int a = 2, b = 3, temp;
        temp = a;
        a = b;
        b = temp;
        printf("Value of a after swapping: %d\n:", a);
        printf("Value of b after swapping: %d\n:", b);
        getch();
}
```

س: کون سے آپریٹرز ارتھمیٹک آپریشنز کے لیے استعمال ہوتے ہیں؟

ارتھمیٹک آپریٹرز (Arithmetic Operators)

ارتھمیٹک آپریٹرز ڈیٹا پر ریاضی کے کام کرنے کے لیے استعمال ہوتے ہیں۔ مندرجہ ذیل جدول ارتھمیٹک آپریٹرز کو ان کی تفصیل کے ساتھ پیش کرتا ہے۔

آپریٹر	نام	تفصیل
/	تقسیم کا آپریٹر	یہ بائیں طرف کی قیمت کو دائیں طرف کی قیمت سے تقسیم کرنے کے لیے استعمال ہوتا ہے۔
*	ضرب کا آپریٹر	یہ دو قیمتوں کو ضرب دینے کے لیے استعمال ہوتا ہے۔
+	جمع کا آپریٹر	یہ دو قیمتوں کو جمع کرنے کے لیے استعمال ہوتا ہے۔
-	تفریق کا آپریٹر	یہ بائیں طرف کی قیمت کو دائیں طرف سے تفریق کرتا ہے۔
%	ماڈولس کا آپریٹر	یہ بائیں طرف کی قیمت کو دائیں طرف کی قیمت پر عددی تقسیم کرنے کے بعد بچنے والی قیمت دیتا ہے۔

taleemcity.com

تقسیم (Division)

تقسیم کا آپریٹر (/) بائیں اوپرینڈ کی قیمت کو دائیں اوپرینڈ کی قیمت پر تقسیم کرتا ہے۔ جیسے: **float result = 3.0 / 2.0;** سٹیٹمنٹ چلنے کے بعد متغیر کا نتیجہ 1.5 آتا ہے۔ اگر دونوں اوپرینڈ انٹیجر ٹائپ ہیں تو تقسیم کا نتیجہ بھی int ٹائپ کا ہو گا۔ انٹیجر کا جواب دینے کے لیے باقی نظرانداز کر دیا جاتا ہے۔ کوڈ کی مندرجہ ذیل لائن پر غور کریں۔ **float result = 3 / 2;**

چونکہ دونوں قیمتیں int ٹائپ کی ہیں لہذا جواب بھی ایک انٹیجر ہو گا جو 1 ہے۔ جب یہ قیمت (1) کو فلوٹ ٹائپ کے متغیر result سے منسوب کیا جائے گا تو یہ (1) فلوٹ ٹائپ میں تبدیل ہو جاتا ہے۔ لہذا ویلیو 1.0 کو متغیر result میں محفوظ کیا جاتا ہے۔ اگر ہم درست اور جامع جواب حاصل کرنا چاہتے ہیں تو پھر اوپرینڈ ایک فلوٹ ٹائپ کا ہونا چاہیے۔ کوڈ کی مندرجہ ذیل لائن پر غور کریں: **float result = 3.0 / 2;**

مندرجہ بالا مثال میں متغیر result کی قیمت 1.5 ہے۔

مثال: ایک پروگرام لکھیں جو چاکلیٹ کے ایک ڈبہ اور چاکلیٹ کی تعداد جو اس ڈبہ میں ہے کی ان پٹ لے۔ اور پروگرام ایک چاکلیٹ کی قیمت دکھائے۔

Program

```
#include<stdio.h>
#include<conio.h>
void main()
{
        clrscr();
        float box_price, num_of_chocolates, unit_price;
        printf("Please enter the price of whole box of chocolates:");
        scanf("%f ", &box_price);
        printf("Please enter the number of chocolates in the box:");
        scanf("%f ", &num_of_chocolates);
        unit_price = box_price / num_of_chocolates;
        printf("The price of a single chocolates is: %f ", unit_price);
        getch();
}
```

Output

```
Please enter the price of whole box of chocolates: 150
Please enter the number of chocolates in the box: 50
The price of a single chocolates is: 3.000000
```

ضرب (Multiplication)

ضرب آپریٹر (*) ایک بائنری آپریٹر ہے جو دو نمبروں کو ضرب دیتا ہے۔ جیسے:

int multiply = 5 * 5;

اس سٹیٹمنٹ پر عمل درآمد کے بعد ویری ایبل (multiply) کی قیمت 25 ہو جائے گی۔

مثال: ایک پروگرام لکھیں جو مستطیل کی لمبائی اور چوڑائی بطور ان پٹ لے اور سکرین پر مستطیل کا رقبہ دکھائے۔

Program

```
#include<stdio.h>
#include<conio.h>
void main()
{
        clrscr();
        float length, width, area;
        printf("Please enter the length of rectangle:");
        scanf("%f ", &length);
        printf("Please enter the width of rectangle:");
        scanf("%f ", &width);
        area = length * width;
        printf("Area of rectangle is : %f ", area);
        getch();
}
```

Output

```
Please enter the length of rectangle: 6.5
Please enter the width of rectangle: 3
Area of rectangle is: 19.500000
```

جمع (Addition)

جمع کا آپریٹر (+) دو اوپرینڈ کو جمع کرتا ہے۔ جیسے:

int add = 10 + 10;

متغیر add کی قیمت 20 ہو جائے گی۔

مثال: ایک پروگرام لکھیں جو دو مضامین کے نمبر زیوزر سے ان پٹ لے اور ان نمبروں کو جمع کرکے سکرین پر دکھائے۔

Program

```
#include <stdio.h>
#include<conio.h>
void main()
{
        clrscr();
        int sum, math, science;
        printf("Enter marks of Mathematics:");
        scanf("%d", &math);
        printf("Enter marks of Science:");
        scanf("%d", &science);
        sum = math + science;
        printf("Sum of marks is : %d", sum);
        getch();
}
```

Output

```
Enter marks of Mathematics: 90
Enter marks of Science: 80
Sum of marks is: 170
```

taleemcity.com

سٹیٹمنٹ ;a = a + 1 ویری ایبل a کی قیمت میں 1 کا اضافہ کرنے کے لیے استعمال ہوتی ہے۔ C لینگوئج میں اس سٹیٹمنٹ کو ;++a یا ;a++ بھی لکھا جاسکتا ہے۔ اسی طرح ;--a یا ;a-- ویری ایبل a کی قیمت میں 1 کو کم کرنے کے لیے استعمال کیا جاتا ہے۔

تفریق (Subtraction)

تفریق کا آپریٹر (-) بائیں اوپرینڈ سے دائیں اوپرینڈ کو تفریق کرتا ہے۔ جیسے:

```
int result = 20 – 15;
```

تفریق کرنے کے بعد متغیر result کی قیمت 5 ہو جاتی ہے۔

ماڈولس آپریٹر (Modulus Operator)

ماڈولس آپریٹر (%) بائیں اوپرینڈ کو دائیں اوپرینڈ پر تقسیم کرتا ہے اور تقسیم کے بعد بچنے والی باقی رقم لوٹاتا ہے۔ ماڈولس آپریٹر انٹیجر ڈیٹا ٹائپ پر کام کرتا ہے۔ جیسے:

```
int remaining = 14 % 3;
```

اگر ہم 14 کو 3 سے تقسیم کریں تو 2 باقی بچے گا۔ اس لیے ویری ایبل remaining کی قیمت 2 ہو گی۔

مثال: ایک پروگرام لکھیں جو کسی ان پٹ کیے گے نمبر کا ماڈولس آپریٹر استعمال کرتے ہوئے صحیح عدد معلوم کرے۔

Program

```
#include <stdio.h>
#include<conio.h>
void main()
{
        clrscr();
        int num, digit;
        printf("Enter a number:");
        scanf("%d", &num);
        digit = num % 10;
        printf("Right most digit of number you entered is: %d", digit);
        getch();
}
```

Output

```
Enter a number: 789
Right most digit of number you entered is: 9
```

taleemcity.com

اگرچہ C لینگوئج میں ارتھمیٹک سٹیٹمنٹس لکھتے ہوئے ایک عام غلطی الجبرا کے روزمرہ کے قوانین کو استعمال کرنا ہے۔ مثلاً 6 * y کو 6y لکھنا اور x * x * x کو x^3 وغیرہ۔ اس کے نتیجے میں کمپائلر ایرر (Complier Error) آتا ہے۔

س: ریلیشنل آپریٹرز کیا ہیں؟ ایک مثال کے ساتھ بیان کریں۔

ریلیشنل آپریٹرز (Relation Operator)

ریلیشنل آپریٹرز دو قیمتوں کے درمیان تعلق کا تعین کرنے کے لیے انکا موازنہ کرتے ہیں۔ ریلیشنل آپریٹرز شناخت کرتے ہیں کہ قیمتیں برابر ہیں یا برابر نہیں۔ ایک قیمت دوسری قیمت سے بڑی ہے یا چھوٹی ہے۔ سی لینگوئج ہمیں نیومیرک اور char ٹائپ کے ڈیٹا پر ریلیشنل آپریٹرز انجام دینے کی اجازت دیتی ہے۔ بنیادی ریلیشنل آپریٹرز ان کی تفصیل کے ساتھ ذیل میں دیے گئے ہیں:

ریلیشنل آپریٹر	تفصیل
==	کے برابر ہے
! =	کے برابر نہیں ہے
>	سے زیادرہ
<	سے کم
> =	سے زیادہ یا برابر
< =	سے کم یا برابر

ریلیشنل آپریٹرز دو اوپرینڈ پر آپریشن انجام دیتے ہیں اور بولین ایکسپریشن (صحیح یا غلط) میں نتیجہ دیتے ہیں۔ اگر ایک قیمت صحیح (True) ہو تو قیمت 1 ہو گی اور اگر غلط (False) ہو تو قیمت 0 سے ظاہر کی جاتی ہے۔

ریلیشنل ایکسپریشن	تفصیل	جواب
5 == 5	5 برابر ہے 5 کے؟	True
5 ! = 7	5 برابر نہیں ہے 7 کے؟	True
5 > 7	5 بڑا ہے 7 سے؟	False
5 < 7	5 چھوٹا ہے 7 سے؟	True
5 > = 5	5 بڑا ہے یا برابر ہے 5 کے؟	True
5 < = 4	5 چھوٹا ہے یا برابر ہے 4 کے؟	False

س: اسائنمنٹ آپریٹر (=) اور برابر کے آپریٹر (==) میں کیا فرق ہے؟

اسائنمنٹ آپریٹر (=) اور برابر کے آپریٹر (==) میں فرق

سی لینگوئج میں (==) آپریٹر دو ایکسپریشن کی مساوات کی جانچ کے لیے استعمال ہوتا ہے کہ وہ برابر ہیں کہ نہیں۔ جبکہ (=) آپریٹر دائیں والی ایکسپریشن کو بائیں والے ویری ایبل سے منسوب کرتا ہے۔ ڈبل ایکوئل (equal) آپریٹر (==) چیک کرتا ہے کہ دونوں طرف یہ آپریٹرز برابر ہیں یا نہیں۔ سنگل ایکوئل آپریٹر (=) دائیں اوپرینڈز کو بائیں اوپرینڈز سے منسوب کرتا ہے۔

ہم printf فنکشن کو ریلیشنل ایکسپریشن کا نتیجہ دکھانے کے لیے بھی استعمال کر سکتے ہیں۔ مثلاً

```
printf("%d", 5 == 5); // This statement displays 1
printf("%d", 5 > 7); // This statement displays 0
```

FSC, Matric Notes, Test Papers, Past Papers & other study matrial, available on taleemcity.com

س: منطقی آپریٹرز کیا ہیں؟ مثال کے ساتھ بیان کریں۔

منطقی آپریٹرز (Logical Operators)

منطقی آپریٹرز بولین ایکسپریشن پر آپریشن سرانجام دیتے ہیں اور اس کے نتیجے میں جواب بھی بولین ایکسپریشن ہوتا ہے۔ ریلیشنل آپریشن کا جواب بولین ایکسپریشن ہے۔ ایک سے زیادہ ریلیشنل ایکسپریشن کا جائزہ لینے کے لیے منطقی آپریٹرز کو سرانجام دیا جاسکتا ہے۔ مندرجہ ذیل ٹیبل بنیادی منطقی آپریٹرز اور ان کی تفصیل کو ظاہر کرتا ہے۔

آپریٹر	تفصیل
&&	Logical AND
||	Logical OR
!	Logical NOT

AND آپریٹر (&&)

AND آپریٹر (&&) دو بولین ایکسپریشن کو بطور اوپرینڈ لیتا ہے اور اگر اس کے دونوں ایکسپریشن درست (true) ہوں تو نتیجہ درست آتا ہے۔ اگر کوئی بھی آ ایکسپریشن غلط (false) ہو تو جواب غلط آتا ہے۔ AND آپریٹر کے لیے ٹروتھ ٹیبل ذیل میں دکھایا گیا ہے۔

ایکسپریشن 1	آپریٹر	ایکسپریشن 2	رزلٹ
False	&&	False	False
False	&&	True	False
True	&&	False	False
True	&&	True	True

OR آپریٹر (||)

OR آپریٹر دو بولین ایکسپریشن لیتا ہے اور کم از کم ایک ایکسپریشن درست (true) ہونے پر جواب درست دیتا ہے۔ OR آپریٹر کے لیے ٹروتھ ٹیبل ذیل میں دکھایا گیا ہے:

ایکسپریشن 1	آپریٹر	ایکسپریشن 2	رزلٹ
False	||	False	False
False	||	True	True
True	||	False	True
True	||	True	True

taleemcity.com

NOT آپریٹر (!)

NOT آپریٹر بولین ایکسپریشن کی قیمت کی نفی یا الٹ کرتا ہے۔ یہ اسے درست (true) بنا دیتا ہے اگر یہ غلط ہے اور اگر یہ غلط (false) ہے تو اسے درست بنا دیتا ہے۔ آپریٹر کے لیے ٹروتھ ٹیبل ذیل میں دیا گیا ہے۔

ایکسپریشن	آپریٹر	رزلٹ
False	!	True
True	!	False

منطقی آپریٹرز کی مثالیں:

مندرجہ ذیل ٹیبل مثالوں کی مدد سے منطقی آپریٹرز کے تصور کو ظاہر کرتا ہے۔

منطقی آپریٹرز	تفصیل	جواب
3 < 4 && 7 > 8	3 چھوٹا ہے 4 سے AND 7 بڑا ہے 8 سے ؟	False
3 == 4 || 3 > 1	3 برابر ہے 4 کے OR 3 بڑا ہے 1 سے ؟	True
!(4 > 2 || 2 == 2)	NOT (4 بڑا ہے 2 سے OR 2 برابر ہے 2 کے)؟	False
6 <= 6 && !(1 > 2)	6 چھوٹا یا برابر ہے 6 کے NOT AND (1 بڑا ہے 2 سے)؟	True
8 > 9 || !(1 <= 0)	8 بڑا ہے 9 سے NOT OR (1 چھوٹا ہے یا برابر ہے 0 کے)؟	True

سی لینگویج شارٹ سرکٹ ایویلیوئیشن (Short circuit evaluation) کرتی ہے۔ اس کا مطلب ہے کہ:

taleemcity.com

- اگر AND آپریٹر کو حل کرتے ہوئے ایکسپریشن آپریٹر کی بائیں طرف والا حصہ غلط (False) ہو تو پورا ایکسپریشن حل کیے بغیر جواب غلط ہو جاتا ہے۔
- اگر OR آپریٹر کو حل کرتے ہوئے ایکسپریشن آپریٹر کی بائیں طرف والا حصہ درست (true) ہو تو پورا ایکسپریشن حل کیے بغیر جواب درست ہو جاتا ہے۔

س: یونری آپریٹرز اور بائنری آپریٹرز میں کیا فرق ہے؟

یونری بمقابلہ بائنری آپریٹرز (Unary vs Binary Operator)

تمام آپریٹرز کو دو بنیادی اقسام میں تقسیم کیا جاسکتا ہے جس کی بنیاد پر آپریٹرز کو لاگو کیا جاسکتا ہے۔

یونری آپریٹرز (Unary Operator)

یونری آپریٹرز صرف ایک آپریشن پر لاگو ہوتے ہیں جیسے منطقی NOT(!) آپریٹر کے پاس صرف ایک اوپرینڈ ہوتا ہے۔ سائن آپریٹر (-) یونری آپریٹر کی ایک اور مثال ہے۔ جیسے 5- ۔

بائنری آپریٹرز (Binary Operator)

بائنری آپریٹرز کو آپریشن کرنے کے لیے دو ایکسپریشن کی ضرورت ہوتی ہے۔ مثلاً تمام ارتھمیٹک اور ریلیشنل آپریٹرز بائنری آپریٹرز ہیں۔ منطقی آپریٹرز && اور || بھی بائنری آپریٹرز ہیں۔

ٹرنری آپریٹر (Ternary Operator)

سی پروگرامنگ لینگویج میں ایک ٹرنری آپریٹر (Ternary Operator) ہے جو تین اوپرینڈز پر کام کرتا ہے۔

س: آپریٹرز کی ترجیح سے کیا مراد ہے؟ کون سی آپریٹر کو C زبان میں سب سے زیادہ فوقیت حاصل ہے؟

آپریٹرز کی ترجیح (Operator Precedence)

اگر ایکسپریشن میں ایک سے زیادہ آپریٹرز ہیں تو سوال یہ پیدا ہوتا ہے کہ پہلے کسے حل کریں گے۔ اس مسئلے کو حل کرنے کے لیے ہر آپریٹر کو ترجیح دی گئی ہے۔ جس آپریٹر کی ترجیح زیادہ ہے وہ دوسرے آپریٹر سے پہلے حل ہوگا۔ اگر آپریٹر کی ایک جیسی ترجیح ہو تو دائیں طرف والے آپریٹر سے پہلے بائیں طرف والا آپریٹر حل کریں گے۔

ترجیح	آپریٹر
1	( )
2	!
3	* , / , %
4	+ , -
5	> , < , >= , <=
6	== , !=
7	&&
8	||
9	=

مثال: 18 / 2 * 3 + 7 % 3 + (5 * 4) کو حل کریں۔

= 18 / 2 * 3 + 7 % 3 + (5 * 4);	( )	پہلے بریکٹ حل ہوگی
= 18 / 2 * 3 + 7 % 3 + 20;	/	دوسرے نمبر پر تقسیم کا عمل ہوگا
= 9 * 3 + 7 % 3 + 20;	*	پھر ضرب کا عمل ہوگا
= 27 + 7 % 3 + 20;	%	پھر ماڈولس آپریٹر حل ہوگا
= 27 + 1 + 20;	+	پھر جمع کا عمل ہوگا
= 28 + 20;	+	پھر جمع کا عمل ہوگا
= 48;		

taleemcity.com