

arithmetic_operator.c

```
1 #include <stdio.h>
2 void main()
3 {
4     int a = 9, b = 4, c;
5     c = a+b;
6     printf("a+b = %d \n",c);
7     c = a-b;
8     printf("a-b = %d \n",c);
9     c = a*b;
10    printf("a*b = %d \n",c);
11    c=a/b;
12    printf("a/b = %d \n",c);
13    c=a%b;
14    printf("Remainder when a divided by b = %d \n",c);
15 }
16
17 /*
18 * Output: a=9, b=4, c=0
19 */
20
21 /*
22 * Note : If we use void main instead of int main, return 0 statement is not
required.
23 * However, using int main is considered a better practice as it indicates that the
program has executed successfully.
24 */
```

arithmetic_operator_2.c

```
1 #include <stdio.h>
2 // C program to demonstrate arithmetic operators
3 int main()
4 {
5     int a = 9, b = 4, c;
6     c = a+b;
7     printf("a+b = %d \n",c);
8     c = a-b;
9     printf("a-b = %d \n",c);
10    c = a*b;
11    printf("a*b = %d \n",c);
12    c=a/b;
13    printf("a/b = %d \n",c);
14    c=a%b;
15    printf("Remainder when a divided by b = %d \n",c);
16    return 0;
17 }
18
19 /*
20 * Output:
21 a+b = 13
22 a-b = 5
23 a*b = 36
24 a/b = 2
25 Remainder when a divided by b = 1
26 */
```

relational_operator.c

```
1 #include <stdio.h>
2 // C program to demonstrate relational operators
3 int main()
4 {
5     int a = 5, b = 5, c = 10;
6     printf("%d == %d = %d \n", a, b, a == b); // true
7     printf("%d == %d = %d \n", a, c, a == c); // false
8     printf("%d > %d = %d \n", a, b, a > b); //false
9     printf("%d > %d = %d \n", a, c, a > c); //false
10    printf("%d < %d = %d \n", a, b, a < b); //false
11    printf("%d < %d = %d \n", a, c, a < c); //true
12    printf("%d != %d = %d \n", a, b, a != b); //false
13    printf("%d != %d = %d \n", a, c, a != c); //true
14    printf("%d >= %d = %d \n", a, b, a >= b); //true
15    printf("%d >= %d = %d \n", a, c, a >= c); //false
16    printf("%d <= %d = %d \n", a, b, a <= b); //true
17    printf("%d <= %d = %d \n", a, c, a <= c); //true
18    return 0;
19 }
20
21 /*
22 * Output:
23 5 == 5 = 1
24 5 == 10 = 0
25 5 > 5 = 0
26 5 > 10 = 0
27 5 < 5 = 0
28 5 < 10 = 1
29 5 != 5 = 0
30 5 != 10 = 1
31 5 >= 5 = 1
32 5 >= 10 = 0
33 5 <= 5 = 1
34 5 <= 10 = 1
35 */
```

logical_operator.c

```
1 #include <stdio.h>
2
3 // C program to demonstrate logical operators
4
5 int main (){
6     int a = 5, b = 5, c = 10, result;
7     result = (a == b) && (c > b);
8     printf("(a == b) && (c > b) equals to %d \n", result);
9     result = (a == b) && (c < b);
10    printf("(a == b) && (c < b) equals to %d \n", result);
11    result = (a == b) || (c < b);
12    printf("(a == b) || (c < b) equals to %d \n", result);
13    result = (a != b) || (c < b);
14    printf("(a != b) || (c < b) equals to %d \n", result);
15    result = !(a == b);
16    printf("!(a == b) equals to %d \n", result);
17    result = !(a == b);
18    printf("!(a == b) equals to %d \n", result);
19    return 0;
20 }
21 ****
22 * Output: a=b=5, c=10
23 ****
24 (a == b) && (c > b) equals to 1
25 (a == b) && (c < b) equals to 0
26 (a == b) || (c < b) equals to 1
27 (a != b) || (c < b) equals to 0
28 !(a == b) equals to 1
29 !(a == b) equals to 0
30 */
31 */
```

assignment_operator.c

```
1 #include <stdio.h>
2
3 // C program to demonstrate assignment operators
4
5 int main (){
6
7     // variable declaration
8     int a,b,c;
9
10    // variable initialization
11    a = 5;
12    b = 10;
13    c = 15;
14
15    // using assignment operators
16    a += 5; // equivalent to a = a + 5
17    b -= 5; // equivalent to b = b - 5
18    c *= 2; // equivalent to c = c * 2
19
20    // printing the results
21    printf("After assignment operations:\n");
22    printf("a = %d\n", a);
23    printf("b = %d\n", b);
24    printf("c = %d\n", c);
25
26    return 0;
27 }
28
29 ****
30 * Output: a=5, b=10, c=15
31 ****
32 After assignment operations:
33 a = 10
34 b = 5
35 c = 30
36 */
```

```
1 // C Program to demonstrate the working of assignment operators
2 #include <stdio.h>
3 int main()
4 {
5     int a = 5, c;
6     c = a;
7     printf("c = %d \n", c);
8     c += a; // c = c+a
9     printf("c = %d \n", c);
10    c -= a; // c = c-a
11    printf("c = %d \n", c);
12    c *= a; // c = c*a
13    printf("c = %d \n", c);
14    c /= a; // c = c/a
15    printf("c = %d \n", c);
16    c %= a; // c = c%a
17    printf("c = %d \n", c); // Remainder of c divided by a
18    return 0;
19 }
20
21 ****
22 * Output: a=5, c=5
23 ****
24 c = 5
25 c = 10
26 c = 5
27 c = 25
28 c = 5
29 c = 0
30 */
```