

## Exercise 4 - Expressions

### Objective

The major objective is to practice the operators, expression evaluation and assignment statements.

### Reference Material

This is based on the *Data Types* chapter and the *Expressions* chapter.

This practical session is located in the following directory:

<i>Windows Directory:</i>	<b>c:\qacprg\operator</b>
<i>Linux Directory:</i>	<b>/home/user1/qacprg/OPERATOR</b>

### Overview

The questions are 'pencil and paper' exercises on operators and data types. To check your answers, open the appropriate project in Visual Studio / Linux environment and build / run the program.

### Practical Outline

1. The program includes many of the types and operators encountered so far. What will be displayed by the `printf` functions? To check your answers, open the Visual Studio Solution (See notes at the end of this document) **opers1.sln**, and build / run the program. On Linux use **make**, and build / run the program.

```
#include <stdio.h>
int main (void)
{
    int i = 7, j;

    /* increment operators */
    printf("i started off as %d\t", i);
    printf("++i is %d\t", ++i);
    printf("and (think about this!) i++ is %d\n\n", i++);

    /* assignment operators/updaters */
    j = i;
    printf("Assignment operators ... so\t\t");
    printf("  i = i * 10 is %d\n", i = i * 10);
```

```
printf("but can also be written as:\t\t");
printf("  i *= 10  which is now %d\n", i *= 10);
i = j;
printf("\nalso available are -, / and %% ... so\t");
printf("  i = i %% 7 is %d\n", i = i % 7);
i = j;
printf("but can also be written as:\t\t");
printf("  i %%= 10  which is %d\n", i %= 10);

return 0;
}
```

## Optional

2. This program includes further 'thought-provokers'. Again, what will be 'printf'ed ?

To check your answers, open the Visual Studio Solution **opers2.sln**, and build / run the program. Again, on Linux use **make**, and build / run the program.

```
#include <stdio.h>
int main (void)
{
    int i, j, k;
    int max_val = 17;
    unsigned u;
    long bignumber;
    double db, result;

    /* The unsigned and long integers */

    u = 1;
    bignumber = 100000L;           /* long int constant */

    printf("\nThe unsigned u is %u\n", u);

    printf("The result displayed need not be saved:\n");
    printf("u - 10 is %u\t\t", u - 10);
    printf("u / 2  is %u\n", u / 2);

    printf("\nbignumber starts off as %ld\n", bignumber);
    printf("bignumber *= 100 is %ld\n",
           bignumber *= 100L);
```

```
i = j = k = 0;

i = -8 + max_val - 2;
j = i++;
k = 14 + 1 * 2;
bignumber = 16L / 6L;
printf("\ni is %d\t\t\t", i);
printf("j is %d\n", j);
printf("k is %d\t\t\t", k);
printf("bignumber is %ld\n", bignumber);

/* ... and now a little bit on doubles */

db = 35.05;

result = db * (double) bignumber;
printf("\ndb is %.2f, result is %.2f\n", db, result);

return 0;
}
```