

C PROGRAMING / LINUX [DASL-100]

WEEK 1 [Section 2]

INSTRUCTOR: JEAN CHAGAS VAZ

Thursday, May 25, 2017, 11:03

➤ Program to Compute Quotient and Remainder

➤ This program evaluates the quotient and remainder when an integer is divided by another integer.

```
#include <stdio.h>
int main(){

    int dividend, divisor, quotient, remainder;

    printf("Enter dividend: ");
    scanf("%d", &dividend);

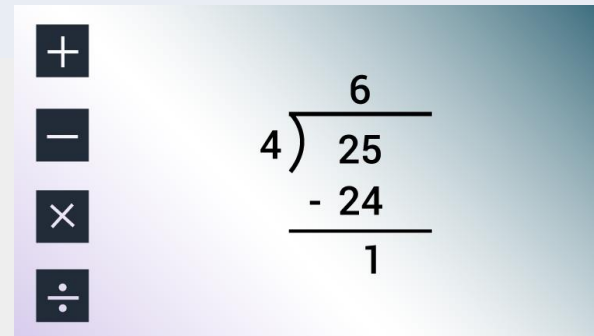
    printf("Enter divisor: ");
    scanf("%d", &divisor);

    // Computes quotient
    quotient = dividend / divisor;

    // Computes remainder
    remainder = dividend % divisor;

    printf("Quotient = %d\n", quotient);
    printf("Remainder = %d", remainder);

    return 0;
}
```


$$\begin{array}{r} 6 \\ 4 \overline{) 25} \\ \underline{- 24} \\ 1 \end{array}$$

Output

```
Enter dividend: 25
Enter divisor: 4
Quotient = 6
Remainder = 1
```

Thursday, May 25, 2017, 11:01

➤ Program to Find the Size of a variable

➤ This program declares 4 variables of type int, float, double and char. Then, the size of each variable is evaluated using sizeof operator.

```
#include <stdio.h>
int main()
{
    int integerType;
    float floatType;
    double doubleType;
    char charType;

    // Sizeof operator is used to evaluate the size of a variable
    printf("Size of int: %ld bytes\n",sizeof(integerType));
    printf("Size of float: %ld bytes\n",sizeof(floatType));
    printf("Size of double: %ld bytes\n",sizeof(doubleType));
    printf("Size of char: %ld byte\n",sizeof(charType));

    return 0;
}
```



Output

```
Size of int: 4 bytes
Size of float: 4 bytes
Size of double: 8 bytes
Size of char: 1 byte
```

Thursday, May 25, 2017, 11:05

➤ Program to Demonstrate the Working of long

➤ The long is a size modifier, indicated by keyword long, that may increase the size of a variable during declaration. This program will demonstrate the working of long keyword.

```
#include <stdio.h>
int main()
{
    int a;
    long b;
    long long c;

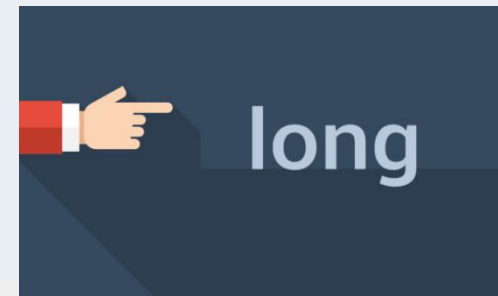
    double e;
    long double f;

    printf("Size of int = %ld bytes \n", sizeof(a));
    printf("Size of long = %ld bytes\n", sizeof(b));
    printf("Size of long long = %ld bytes\n", sizeof(c));

    printf("Size of double = %ld bytes\n", sizeof(e));
    printf("Size of long double = %ld bytes\n", sizeof(f));

    return 0;
}
```

Program to find the size of int, long int



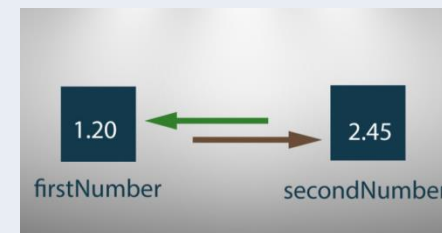
Output

```
Size of int = 4 bytes
Size of long = 8 bytes
Size of long long = 8 bytes
Size of double = 8 bytes
Size of long double = 16 bytes
```

Thursday, May 25, 2017, 11:08

➤ Program to Swap Numbers Using Temporary Variable

➤ This example contains two different techniques to swap numbers in C programming. The first program uses temporary variable to swap numbers, whereas the second program doesn't use temporary variables.



```
#include <stdio.h>
int main()
{
    double firstNumber, secondNumber, temporaryVariable;

    printf("Enter first number: ");
    scanf("%lf", &firstNumber);

    printf("Enter second number: ");
    scanf("%lf",&secondNumber);

    // Value of firstNumber is assigned to temporaryVariable
    temporaryVariable = firstNumber;

    // Value of secondNumber is assigned to firstNumber
    firstNumber = secondNumber;

    // Value of temporaryVariable (which contains the initial value of firstNumber) is assigned to secondNumber
    secondNumber = temporaryVariable;

    printf("\nAfter swapping, firstNumber = %.2lf\n", firstNumber);
    printf("After swapping, secondNumber = %.2lf", secondNumber);

    return 0;
}
```

Output

```
Enter first number: 1.20
Enter second number: 2.45

After swapping, firstNumber = 2.45
After swapping, secondNumber = 1.20
```

Thursday, May 25, 2017, 11:16

➤ Program to Swap Number Without Using Temporary

```
#include <stdio.h>
int main()
{
    double firstNumber, secondNumber;

    printf("Enter first number: ");
    scanf("%lf", &firstNumber);

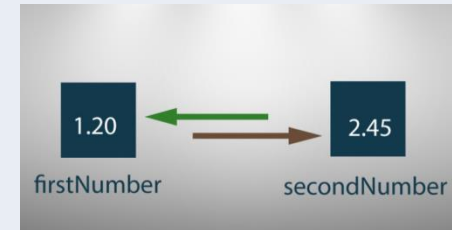
    printf("Enter second number: ");
    scanf("%lf",&secondNumber);

    // Swapping process

    firstNumber = firstNumber - secondNumber;
    secondNumber = firstNumber + secondNumber;
    firstNumber = secondNumber - firstNumber;

    printf("\nAfter swapping, firstNumber = %.2lf\n", firstNumber);
    printf("After swapping, secondNumber = %.2lf", secondNumber);

    return 0;
}
```



Output

```
Enter first number: 10.25
Enter second number: -12.5

After swapping, firstNumber = -12.50
After swapping, secondNumber = 10.25
```

Thursday, May 25, 2017, 11:19

➤ Program to Check Even or Odd

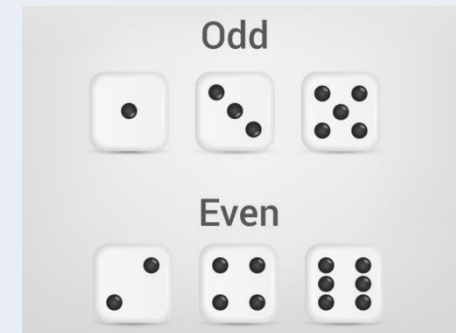
➤ In this example, if...else statement is used to check whether a number entered by the user is even or odd.

```
#include <stdio.h>
int main()
{
    int number;

    printf("Enter an integer: ");
    scanf("%d", &number);

    // True if the number is perfectly divisible by 2
    if(number % 2 == 0)
        printf("%d is even.", number);
    else
        printf("%d is odd.", number);

    return 0;
}
```



Output

```
Enter an integer: -7
-7 is odd.
```



Thursday, May 25, 2017, 11:26

➤ To do List

- Finish Homework 1 (**DUE one day before NEXT SECTION**)
- Repeat Homework 1 “Problem 2” in C++ language. Then highlight the code differences by using commentaries (in the code)(**DUE NEXT SECTION pdf format**)
- Create a Program to Check Odd or Even Using Conditional Operator. (Optional)
- Create a Program to Swap Number Without Using Temporary Variables. (Optional)