

## MTH6150

### Numerical Computing in C and C++

#### Exercise Sheet 3

---

1. Try the example in exercise sheet 2, question 13 if you haven't done so.  
Then, using `int` datatypes for `m` and `n`, modify the code to use each of the three types of explicit conversions listed in lecture 3, slide 9. For example, to use a C++-style cast, you would replace `m/n` with `static_cast<double>(m)/static_cast<double>(n)`
2. Run the code on lecture 3, slide 12, and add code to output `k` and `y`. Is there any warning when compiling (i.e building) the code? Does the code run and produce the expected output.
3. Check that you understand what the modulus and quotient are by running the code on slide 13, using various values for `a` and `n`, and outputting `q` and `r`.
4. Write code to declare a variable of type `double`, then input its value using `cin`, calculate the square of the variable and output the result.

After compiling the code, run it several times without recompiling, inputting a different number each time.

5. Write code to declare an `int` variable `n`, then input its value using `cin`.

Then, if the `n` is even, run the following command:

```
cout << n << " is even" << endl;
```

otherwise output that `n` is odd.

To check if an integer `n` is even, use

```
if(n%2==0)
```

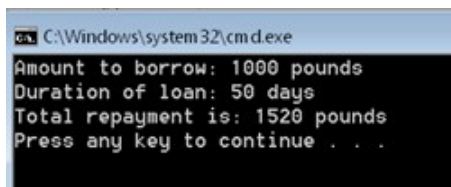
I.e. the remainder of `n` modulus 2 is 0  $\Leftrightarrow$  `n` is even.

6. Change the code on slide 24 so that `k` is input using `cin`, then run it with various inputs to check that it behaves as expected.
7. A pay-day loan company charges simple interest at 1% per day, plus a one-off fee of £20. Write a program that (by outputting a message to the screen) asks the user to enter an amount they want to borrow, then when they have done that, asks the user to enter the number of days they want to borrow for. Finally, the program should output the total amount they will need to repay.

The amount to be repaid  $T$  is calculated as

$$T = B \left( 1 + \frac{rd}{100} \right) + 20$$

where  $d$  is the length of the loan in days and  $r$  is the daily simple interest rate.



Credit companies legally have to state the annual percentage rate (APR), which is defined as  $a$  in

$$T = \left(1 + \frac{a}{100}\right)^y B$$

where  $B$  is the initial amount borrowed,  $T$  is the total amount to be repaid and  $y$  is the length of the loan in years.

Rearrange this formula to find  $a$ , and modify the program to also output the APR. The function `pow` in the `<cmath>` library will be helpful here. In the following code,  $z = x^y$ .

```
#include <cmath>
...
double x = 3.0;
double y = 2.0;
double z = pow(x, y);
```

8. The numerical inputs in question 7 must be non-negative. Add code to check that this is the case, and make the program output an appropriate message if a negative number is input, instead of the usual output.

There may also be input that is not a valid number. If the input (or at least the first characters) cannot be put into the variable that will hold it, e.g. if we enter letters where a number is expected, then the input stream (which is called `cin`) will be left in an invalid state. We can check this by calling `cin.good()` (we haven't covered this type of code yet, but you can just use it as below). If the input is OK, `cin.good()` is `true`, otherwise it is `false`. One way to organize the code is like this, after inputting the numbers:

```
if(!cin.good()){
    // output error message
}
else if(...){ // here put a check for negative numbers
    // output a different error message
}
else{
    // proceed with valid input
}
```

Experiment with this, entering invalid input such as text or mixtures of numbers and text to see how it is dealt with.