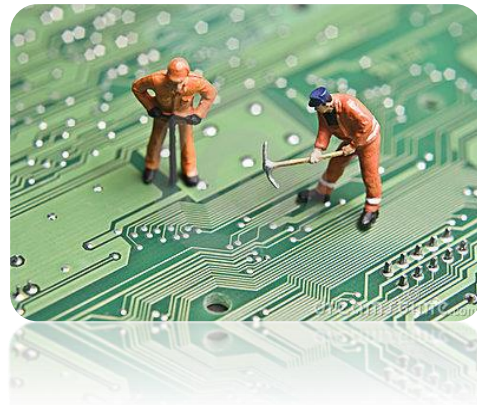


Computational **Problem** **Solving** (CS100)

*Shafay Shamail, Arif Zaman and
Safae Ullah Chaudhary*





Lab 6 Objectives:

- Practice Loops and nested loops.

Lab Guidelines

1. Make sure you get your work graded before the lab time ends.
2. You put all your work onto the LMS folder designated for the lab (i.e. "Lab06") before the time the lab ends.
3. Talking to each other is NOT permitted. If you have a question, ask the lab assistants.
4. If you are a hot-shot C++ expert, you are still not allowed to use any feature of C++ that has not been covered in class or in the lab.
5. The object is not simply to get the job done, but to get it done in the way that is asked for in the lab.



Task 1:

You're working at Target Corporation. Your job as the IT specialist is to implement some code in C++ that takes in the price of an item, multiplies it by the quantity and adds it to the Total amount due.

The code will keep asking for item prices and quantities and will only exit when -1 is entered.

For example, the output from your program should look like this (Note that the numbers the user typed in are colored as 100):

```
Enter item price (-1 to exit): 100
Enter item quantity: 4

Enter item price (-1 to exit): 20
Enter item quantity: 3

Enter item price (-1 to exit): -1
Total amount due is: Rs. 460
```

Bonus:

Modify the above code to give a 10% discount on the total amount if it is over 5000.

Modify the above code to reject negative prices (other than -1) and negative quantities.

STOP AND SHOW YOUR WORK TO THE TA

Previously, we saw how to use `cin` to get input from the user. But `cin` is programmed in such a manner that it considers two inputs separated by a space as two different variables. For instance:

```
1  string s;  
2  cin >> s;  
3  cout << s << s << endl;
```

If the user enters `Hello World`,
Line 2 will cause only `Hello` to be stored in the variable `s`.
The entire line `Hello World` is not stored, because there is space between `Hello` and `World`.
Line 3 will cause `HelloHello` to be printed.

If we want to get a complete line as input, we need to use the `getline` function as shown below

```
1  string s;  
2  getline(cin, s);  
3  cout << s << s;
```

Here, if the user enters `Hello World`,
This program reads the entire line typed by the user and stores it in the string variable called `myline`.
The output of line 3 would then be `Hello WorldHello World`.

Task 2:

In this task you will implement a digital version of the talk back parrot.
Your program should repeatedly ask the user for a string input.
Whatever the user types, your program should repeat exactly the same thing.
However your parrot should say "Sorry" and exit when the user enters 'Stop it'.

A sample run of the program is shown below:

```
Say something:  
Hi, how are you?  
Hi, how are you?  
  
Say something:  
I am fine thanks.  
I am fine thanks.  
  
Say something:  
You are stupid.  
You are stupid.  
  
Say something:  
Stop it  
Sorry
```

STOP AND SHOW YOUR WORK TO THE TA



Task 3

In the previous lab, we had learnt that you can get the digits of a two-digit number by:

```
right_digit = number % 10;
left_digit = number / 10;
```

Try to see what happens for different numbers if the number has more than two digits, like four or seven digits. Remember `ints` can go up to about 2,000,000,000 or so.

Using this insight, make code that prints out the digits of a number starting from the right most digit. For example if the user types 5786, the computer should type back 6, 8, 7, 5.

STOP AND SHOW YOUR WORK TO THE TA

Task 4

A number is called an Armstrong number if the sum of the cubes of its individual digits is equal to the number itself. For example 371 is an Armstrong number as $(3^3) + (7^3) + (1^3) = 371$. Some other Armstrong numbers are: 0, 1, 153, 370, 407.

1. Write a C++ code which finds whether a number is an Armstrong number or not.
2. Finally write your code so that it repeatedly asks for a number, then answers if it is an Armstrong number or not. Your program should stop only when the user types a zero (after first answering the fact that 0 is an Armstrong number).

Note that to compute the cube of a number, you could do:

```
int main()
{
    int n, cube;
    cin >> n;
    cube = n*n*n;
```

Or you could use the `pow` function to compute the third power of a number by:

```
#include <cmath>
int main()
{
    int n, cube;
    cin >> n;
    cube = pow(n, 3);
```

Hint: You will need to use the code you created in Task 3, to do this effectively.

Hint: You will probably need to have nested loops to do the second part of this task.

Skeleton code is available with the TA however you'll have to trade 10 marks of your lab for the skeleton code.