
Lecture Summary

Data types

Recall that a *data type* is set of values and set of operations on those values. It determines the possible values that can be stored in a variable and the operations that can be performed on that variable. It also determines the amount of memory required to store the value and the way the value is stored in the memory. Some commonly used data types in C++ are following¹:

- **int** - stores integers (whole numbers), without decimals, such as 123 or -123. The `int` data type stores values that are within a range of -2,147,483,648 to 2,147,483,647 (32-bit signed integer).

Other useful data types for integers are: `short` (stores 16-bit integers), `long` (stores 32-bit integers), `long long` (stores 64-bit integers).

- **double** - stores floating point numbers, with decimals, such as 19.99 or -19.99. The `double` data type is a double-precision 64-bit floating point.

Other useful data types for floating point numbers are: `float` (stores 32-bit floating point numbers), `long double` (stores 80-bit floating point numbers).

- **bool** - stores values with two states: `true` or `false`

- **char** - stores single characters, such as 'a' or 'B'. The `char` values are surrounded by single quotes.

The `char` is a type of integer, so it also stores numbers, which are interpreted as characters according to the ASCII encoding. For example, the ASCII value for the character 'a' is 97 and the ASCII value for the character 'b' is 98.

- **std::string** - stores text, such as "Hello World". The `std::string` values are surrounded by double quotes.

¹The sizes of these data types may vary from one compiler to another.

Some functions from the C++ library needed for the lab

function	description	example
<code>fmin()</code>	returns the minimum of two numbers	<code>int m = fmin(3, 5);</code> <code>// 3 is stored in m</code>
<code>fmax()</code>	returns the maximum of two numbers	<code>int m = fmax(3, 5);</code> <code>// 5 is stored in m</code>
<code>abs()</code>	returns the absolute value of a number	<code>int x = abs(-10);</code> <code>// 10 is stored in x</code>
<code>pow()</code>	returns the first argument raised to the power of the second argument	<code>double y = pow(2, 3)</code>
<code>sqrt()</code>	returns the square root of a number	<code>double s =</code> <code>sqrt(16);</code>
<code>ceil()</code>	returns the smallest integer that is greater than or equal to the argument	<code>int z = ceil(9.2);</code>
<code>floor()</code>	returns the largest integer that is less than or equal to the argument	<code>int z = floor(9.2);</code>
<code>lround()</code>	returns the long integer that is closest to the argument	<code>long z =</code> <code>lround(9.2);</code>
<code>rand()</code>	returns a random number	<code>int r = rand();</code>
<code>srand()</code>	sets the seed for the random number generator	<code>srand(time(0));</code>
<code>to_string</code>	Convert to string	<code>std::to_string(10)</code>
<code>stoi(string)</code>	Convert string to int	<code>stoi(s)</code>

To use the above arithmetic functions, you need to include the following header file in your program:

```
#include <cmath>
```

For the last random functions (`rand()` and `srand()`), you need to include the following header file in your program:

```
#include <cstdlib>
```

For String, you need to include the following header file in your program:

```
#include <string>
```

Lab Questions

1. Develop a program `temperature.cpp` that takes the temperature in Fahrenheit as input and converts it to Celsius using the formula: $\text{Celsius} = (\text{Fahrenheit} - 32) * 5 / 9$. Output the converted temperature.
2. Write a program that requests the user's name as input and then asks for their age. Display a message that includes their name and age, like: Hello [Name], you are [Age] years old.
3. Check the voting eligibility: Initialize an integer variable `age` and take input for the user. If the user age is 18 or above print "eligible to vote" else print "not eligible". Use ternary operator for this task.
4. Take input number from user and print to console whether the number is negative or positive. Assuming that zero value will not be given as input.
5. Create a program that asks the user to enter their weight in kilograms and height in meters, then calculates and displays their Body Mass Index (BMI) using the formula: $\text{BMI} = \text{Weight} / (\text{Height} * \text{Height})$.
6. Write a program that takes three int values from the user and prints them in ascending order.

Note: Do not use if statement (we haven't covered it in lectures yet). Use `fmin()` and `fmax()` from the `<cmath>` library.

7. Initialize two string literals and take numeric input for them from the user. Concatenate the strings first and print to console, then convert the numbers from string form to int and perform addition on the numbers and print to console. You can convert string variables to int by using `stoi(string)`.
8. The International Standard Book Number (ISBN) is a 10-digit code that uniquely identifies a book. The rightmost digit of an ISBN, called the checksum digit (denoted as `d10`), is used to verify the accuracy of the other nine digits. The checksum digit is calculated in such a way that the weighted sum of all 10 digits is a multiple of 11.

The weighted sum is calculated using the following formula:

$$1*d_1 + 2*d_2 + 3*d_3 + \dots + 10*d_{10}$$

Here, `di` represents the digit at the i-th position from the left (where `d1` is the first digit, `d2` is the second digit, and so on).

The checksum digit (`d10`) can be any value from 0 to 10. If the value of `d10` is 10, it is conventionally represented by the letter 'X' in an ISBN.

Example: Suppose the first 9 digits of the ISBN are `020131452`. To find the checksum digit `d10`, we compute the weighted sum:

$$1*0 + 2*2 + 3*0 + 4*1 + 5*3 + 6*1 + 7*4 + 8*5 + 9*2 = 115$$

Now, we need to find a `d10` such that: $115 + 10*d_{10}$ is a multiple of 11.

The only value of `d10` that satisfies this condition is `5`, because $115 + 10*5 = 165$, and 165 is a multiple of 11. Thus, the full 10-digit ISBN is `0201314525`.

Task:

Write a program that takes a 9-digit integer as input, computes the checksum digit `d₁₀` and prints the complete 10-digit ISBN number. If the checksum digit `d₁₀` is 10, simply print the letter 'X'.

Hint: You can access individual digits of a 9-digit integer `x` using the modulus (`%`) and division (`/`) operations as shown below:

```
int d9 = x % 10;
```

```
int d8 = (x / 10) % 10;
```

```
int d7 = (x / 100) % 10;
```

```
// and so on...
```

```
C:\Users\mrkhowaja\Desktop\programs>.\a
Enter the 9-digit ISBN number: 020131454
The 10-digit ISBN number is: 0201314541
```

```
C:\Users\mrkhowaja\Desktop\programs>.\a
Enter the 9-digit ISBN number: 020131452
The 10-digit ISBN number is: 0201314525
```

```
C:\Users\mrkhowaja\Desktop\programs>.\a
Enter the 9-digit ISBN number: 020131455
The 10-digit ISBN number is: 020131455X
```