Laboratorium1 – podstawowe różnice między C a C++

Pierwszy program.

```
#include <iostream>

using namespace std;

int main(void)
{
	cout << "Witaj C++" << endl; // drukuje napis
	return 0;
}
```

Zadanie 1

Stwórz projekt. Przepisz i uruchom powyższy program. Program uruchom również w debuggerze i wykonaj go krok po kroku.

Wejście wyjście

W C++ do obsługi wejścia służą strumienie. Dwa najbardziej podstawowe to `cin` pobierania danych ze standardowego wejścia oraz `cout` do wyświetlania danych na standardowym wyjściu. Kolejnym często używanym strumieniem jest `cerr` służący do wysyłania komunikatów na standardowe wyjście błędów. Wymienione strumienie zdefiniowane są w pliku nagłówkowym `iostream`.

Dane wczytuje się w następujący sposób

```
int a, b, c;
cin >> a;
cin >> b >> c;
```

Wyświetlanie danych odbywa się analogicznie. Manipulator `endl` wysyła znak końca linii do strumienia.

```
cout << "a= " << a << endl;
cout << "b= " << b << " c= " << c << endl;
```

Podobnie jak w przypadku funkcji `printf()/scanf()` z języka C również w przypadku strumieni można określać format wyświetlanych/wczytywanych danych. Służą do tego manipulatory zdefiniowane w pliku nagłówkowym `ios` oraz `iomanip`.

Manipulator	Działanie
setprecision(precyzja)	ustawia precyzję (liczbę miejsc po przecinku) wyświetlanych/wczytywanych liczb
setfill(znak)	Zmienia znak wypełniający
setw(szerokosc)	Ustawia szerokość następnego pola
left	Ustawia wyrównanie do lewej
right	Ustawia wyrównanie do prawej
dec	Zmienia podstawę systemu liczbowego na 10
hex	Zmienia podstawę systemu liczbowego na 16

oct	Zmienia podstawę systemu liczbowego na 8
setbase(podstawa)	Zmienia podstawę systemu liczbowego na podstawa. Zero powoduje, że podstawa będzie rozpoznawana automatycznie na podstawie formatu (0, 0x, itd.)
showbase	Włącza wyświetlanie podstawy systemu liczbowego
noshowbase	Wyłącza wyświetlanie podstawy systemu liczbowego
fixed	Włącza stałą liczbę cyfr po przecinku dla liczb zmiennoprzecinkowych
scientific	Włącza reprezentację naukową dla liczb zmiennoprzecinkowych
resetioflags(flagi)	Zeruje podaną flagę np. ios_base::scientific

Użycie manipulatorów jest bardzo proste. Wystarczy wysłać je do strumienia. Przykład poniżej powoduje wyświetlenie liczby 114 w systemie szesnastkowym i w formacie pozwalającym określić podstawę systemu liczbowego.

```
cout << showbase << hex << 114 << endl;
```

Zadanie 2

Napisz konwerter liczb całkowitych pozwalający na konwersję między zapisem w systemie ósemkowym, dziesiętnym i szesnastkowym. Konwerter powinien pozwalać na wybór reprezentacji danych wprowadzanych i wyświetlanych. Powinna również istnieć możliwość wyboru automatycznego rozpoznawania formatu wprowadzanej liczby.

Typ string

Jednym z mankamentów języka C był brak typu łańcuchowego. C++ uzupełnia ten brak za pomocą typu (i klasy) `string`. Jest on w pełni kompatybilny ze strumieniami.

```
string napis;
cout << "Podaj napis: " << endl;
cin >> napis;
cout << "Napis to: " << napis << ", jego długość to: " << napis.length()
            << ", a pierwszy znak to: " << napis[0] << endl;
```

Zadanie 3

Napisać program wczytujący dane pracowników do tablicy struktur. Struktura zawiera następujące dane: imię, nazwisko, stawkę godzinową, liczbę godzin pracy, datę zatrudnienia. Po wczytaniu program wyświetla dane w postaci tabeli (wyjście programu należy sformatować tak aby kolumny tabeli były wyrównane

Strumienie

Oprócz standardowych strumieni `cin` i `cout` można tworzyć strumienie powiązane z plikami. Zdefiniowane są w pliku nagłówkowym `fstream`. Ich użycie jest analogiczne do strumieni `cin/cout`. Należy tylko pamiętać o zamknięciu pliku gdy już nie jest potrzebny.

```
//zapis do pliku
```

```
ofstream plik_wy("plik.txt");
for (int i = 0; i < 10; i++)
      plik_wy << "Linia: " << i << endl;
plik_wy.close();

//odczyt z pliku
ifstream plik_we("plik.txt");
string linia;
if (plik_we.is_open())
{
      while (!plik_we.eof())
      {
            getline(plik_we, linia);
            cout << linia << endl;
      }
      plik_we.close();
}
```

Pracą strumieni można sterować nie tylko za pomocą manipulatorów ale również za pomocą wbudowanych w nie metod i funkcji niebędących metodami. W powyższym przykładzie warto zwrócić uwagę na metody: `is_open()` (informującą czy plik jest otwarty), `eof()` (informującą czy osiągnięto koniec pliku), `close()` (zamykającą plik) oraz `getline()` (odczytującą całą linię z pliku). Warto również wiedzieć, że za pomocą metody `imbue()` strumieni można zmieniać ustawienia regionalne dla wczytywanych/zapisywanych/wyświetlanych danych.

```
strumien.imbue(std::locale("en_US.utf8"));
```

Istnieje również możliwość tworzenia strumieni związanych z łańcuchami. Pozwalają one na odczyt i zapis. Można np. z ich pomocą uzyskać łańcuch sformatowany za pomocą manipulatorów (utworzony łańcuch można odczytać za pomocą metody `str()`) a także przetwarzać łańcuchy.

```
stringstream strumien;
strumien << "10 20 30" << " napis";
cout<<strumien.str()<<endl;
int a, b, c;
strumien >> a >> b >> c;
cout << a << " " << b << " " << c << endl;
string napis;
strumien >> napis;
cout << napis << endl;
```

Zadanie 4

Zmodyfikuj program z zadania 3 (pracownicy). Tak aby dane wczytywał z pliku tekstowego. Dane liczbowe (wartości rzeczywiste powinny być wyświetlane w formacie polskim i amerykańskim).

Dodatkowe słowa kluczowe - operatory

Nie wszystkie operatory języka C są dla wszystkich jasne i czytelne, w C++ dodano nowe słowa kluczowe, które są mogą niektórym bardziej odpowiadać. Działają identycznie jak stare operatory. I tak: `&&` ↔ `and`, `||` ↔ `or`, `!` ↔ `not`, `!=` ↔ `not_eq`, `&` ↔ `bitand`, `|` ↔ `bitor`, `^` ↔ `xor`, `&=` ↔ `and_eq`, `|=` ↔ `or_eq`, `^=` ↔ `xor_eq`.

Zadanie 5

Napisz program, który wyświetla podaną liczbę całkowitą w systemie dwójkowym.