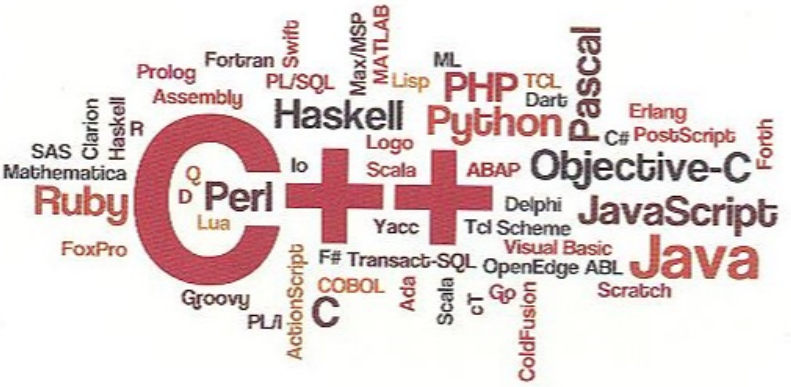
14A. Podstawy pracy w środowisku C++

- Wprowadzenie do programowania w języku C++
- Wzorzec programu
- Definiowanie prostych funkcji

ZADANIE NA START

Programowanie w C++

C++ to jeden z najbardziej znanych języków na świecie. Można w nim pisać aplikacje, które powinny szybko działać. Dlatego język ten jest wykorzystywany m.in. do tworzenia systemów operacyjnych. Znajdź trzy aplikacje napisane w języku C++.



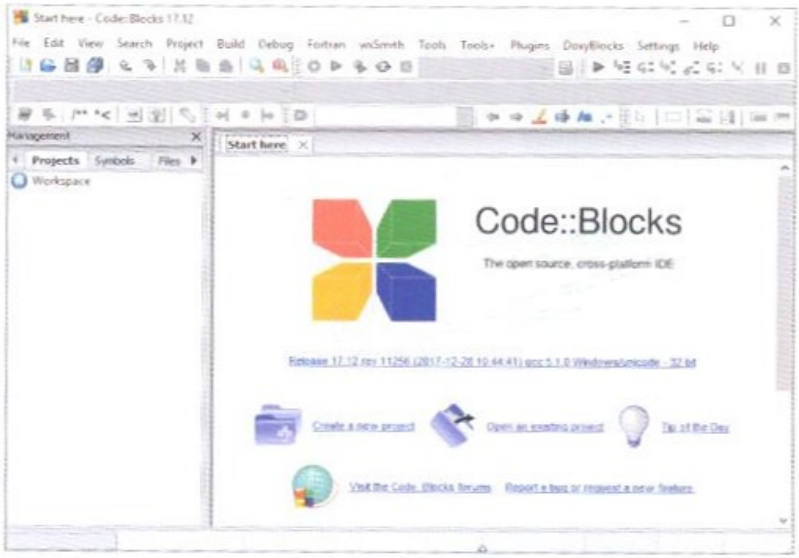
ABC JĘZYKA C++

C++ to język wieloparadygmatowy, co oznacza, że można w nim jednocześnie stosować różne style programowania, np. dzielenie kodu na procedury wykonujące określone operacje, definiowanie programów za pomocą obiektów łączących dane i metody czy programowanie uogólnione, bez znajomości typów danych, na których kod będzie pracował.

POSZUKAJ W INTERNECIE

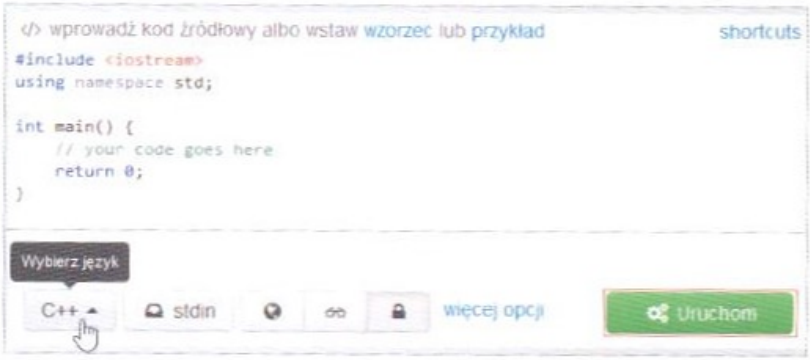
Poszukaj w internecie wywiadów z twórcą języka C++, Bjarnem Stroustrupem, aby dowiedzieć się więcej o nim i jego filozofii.

Kod programu napisanego w języku C++ zapisuje się jako plik tekstowy. Zwyczajowo podczas zapisywania pliku nadaje mu się rozszerzenie CPP i za pomocą kompilatora tworzy plik wykonywalny. Do pisania programów wygodnie jest używać zintegrowanego środowiska deweloperskiego (IDE) zawierającego różne funkcje. W systemach Windows, Linux lub OS pracę w środowisku C++ można rozpocząć od zainstalowania na komputerze środowiska Code::Blocks, które należy pobrać z serwisu **www.codeblocks.org**.



Rys. 1. Zintegrowane środowisko programistyczne Code::Blocks

W ostatnich latach coraz więcej osób nie instaluje środowiska C++ na komputerze, tylko korzysta z niego w chmurze. Programowanie online umożliwia m.in. serwis **ideone.com**, opracowany przez polską firmę z Trójmiasta. Serwis ten obsługuje ponad 40 języków programowania. Warto założyć w nim konto i ustawić domyślny język programowania, język interfejsu oraz włączyć podświetlanie składni, a także używać notatek do opisu programu i etykiet do organizacji swoich kodów. Wszystkie pliki zapisywane są w chmurze, ale można je pobrać na dysk i skorzystać z nich ponownie.

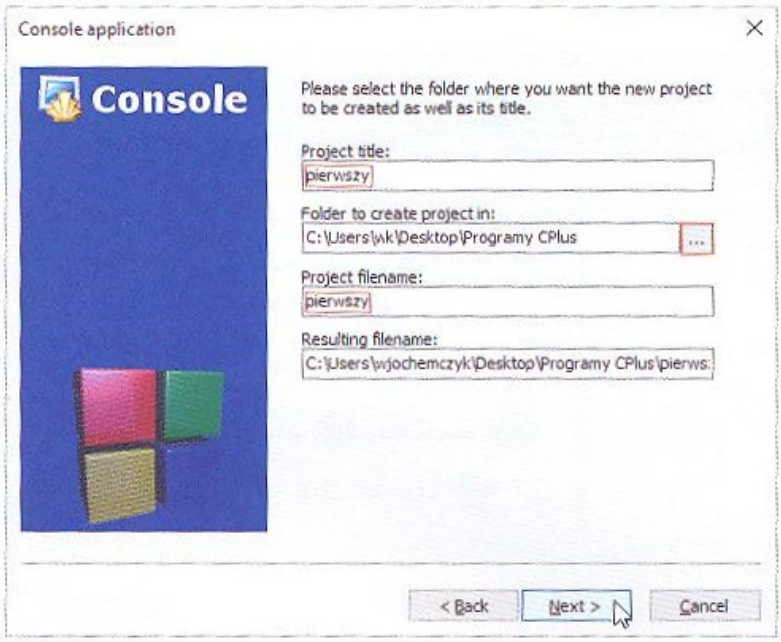


Rys. 2. Wzorzec programu w serwisie **ideone.com**. Kursor wskazuje wybór języka, a czerwona ramka – uruchomienie programu

PRACA W EDYTORZE

Praca w edytorze kodu polega na pisaniu funkcji zgodnie z obowiązującymi regułami. W dalszej kolejności kod jest kompilowany, czyli tłumaczony na język zrozumiały przez komputer, a następnie wykonywany przez komputer. Każda zmiana kodu wymaga ponownej kompilacji.

Aby rozpocząć pracę w edytorze Code::Blocks, należy kliknąć **Create a new project** w oknie programu i wybrać **Console application**. Otworzy się okno, w którym należy wskazać miejsce zapisu projektu, wpisać jego tytuł i nazwę pliku na dysku.



Rys. 3. Tworzenie nowego projektu

Po zapisaniu kodu program uruchamia się za pomocą przycisku ▶ (Run). Aby rozpocząć pracę w edytorze w serwisie **ideone.com**, wystarczy w przeglądarce wpisać adres serwisu i po wgraniu strony można zacząć działać. Aby uruchomić program, należy wybrać opcję **Uruchom**. Wszystkie zapisane programy znajdują się w sekcji **Moje kody**. Każdy z nich można ponownie otworzyć, duplikować lub pobrać na dysk.

Wzorzec programu zawiera: odwołanie do biblioteki **<iostream>**, służącej do obsługi strumieni (dzięki temu można wyświetlać dane na standardowym wyjściu – ekranie), określenie przestrzeni nazw oraz funkcję główną **main()**. Nazwa funkcji jest poprzedzona określeniem typu wyniku – **int** oznacza liczbę całkowitą. Treść funkcji umieszcza się w nawiasach klamrowych. Polecenie **return 0** oznacza zakończenie działania funkcji bez błędu.

```
1. #include <iostream>
2. using namespace std;
3.
4. int main() {
5.     // tu pisze się kod programu
6.     return 0;
7. }
```

Rys. 4. Wzorzec programu

WIEM WIĘCEJ

W C++ jednowierszowy komentarz rozpoczyna się od dwóch ukośników //. Komentarze wielowierszowe zawarte są między znakami /* i */.

ZAPAMIĘTAJ!

Oprócz nawiasów klamrowych warto dodatkowo stosować wcięcia – kod jest wtedy bardziej przejrzysty i zrozumiały dla osoby czytającej.

Jeśli do zapisu wkradnie się błąd składni, pojawi się komunikat o błędzie kompilacji.

```
Błąd kompilacji #stdin błąd kompilacji #stdout 0s 15232KB
prog.cpp: In function 'int main()':
prog.cpp:5:36: error: no match for 'operator<'
  cout << "abrakadabra hokus-pokus" < endl;
  ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~^~~~~~
```

Rys. 5. Przykładowy komunikat w serwisie **ideone.com**

POLECENIE COUT

Instrukcja **cout** wyświetla to, co zostanie napisane po symbolu **<<**, np. tekst (ciąg znaków), liczbę albo wartość zmiennej. Znak **endl** oznacza przejście do nowego wiersza. Każde polecenie kończy się średnikiem.

ZAPAMIĘTAJ!

```
cout << "<ciąg znaków>" << endl;
cout << <liczba> << endl;
cout << <zmienna> << endl;
```

Aby dodać pusty wiersz, można skorzystać z polecenia **cout << endl;**

OPERATORY

Do podstawowych **operatorów arytmetycznych** należą: dodawanie (+), odejmowanie (–), mnożenie (*), dzielenie (/) oraz reszta z dzielenia całkowitego (%).

```
1. #include <iostream>
2. using namespace std;
3.
4. int main() {
5.     cout << 34 + 2 << endl;          36
6.     cout << 34 / 2 << endl;    →     17
7.     cout << 34 % 3 << endl;           1
8.     return 0;
9. }
```

Rys. 6. Wybrane obliczenia w C++

Ćwiczenie 1. Kupowanie obcej waluty

Oblicz, ile musisz zapłacić w złotówkach za 126 euro (aktualny kurs wymiany sprawdź na stronie NBP).

- Zapisz odpowiedni wzór – przemnóż daną kwotę przez aktualny kurs wymiany (pamiętaj o zastąpieniu przecinka kropką). Na przykład gdy 1 euro = 4,20 zł, kod przyjmuje poniższą postać:

```
cout << 126 * 4.2 << endl;
```

- Skompiluj i uruchom program, aby sprawdzić poprawność wyniku.

```
1. #include <iostream>
2. using namespace std;
3.
4. int main() {
5.     cout << 126 * 4.2 << endl;
6.     return 0;
7. }
```

Operatory porównania wykorzystywane w języku C++ zaprezentowano w poniższej tabeli wraz z przykładami.

Operatory porównania			
==	równe	(3 == 3) → True	(4 == 3) → False
!=	różne	(3 != 2) → True	(3 != 3) → False
<	mniejsze	(3 < 4) → True	(4 < 3) → False
<=	mniejsze równe	(4 <= 4) → True	(4 <= 3) → False
>	większe	(4 > 3) → True	(3 > 4) → False
>=	większe równe	(4 >= 4) → True	(4 >= 5) → False

Tab. 1. Operatory porównania i przykłady

Wynikami porównań są wartości logiczne **prawda** lub **fałsz**. Należy pamiętać, że w porównaniu znak równości w C++ jest podwójny. Pojedynczy znak = to operator przypisania, za pomocą którego przypisuje się zmiennej pewną wartość.

Ćwiczenie 2. Porównanie cen wyrażonych w różnych walutach

Porównaj ceny tych samych butów, które w Grecji kosztują 49 euro, a w Polsce 200 zł (aktualny kurs wymiany sprawdź na stronie NBP).

- **Sposób 1** – przelicz cenę butów w Grecji na złotówki, a potem porównaj obie wartości.
- **Sposób 2** – zapisz oba działania w jednym wierszu z wykorzystaniem operatorów arytmetycznych i porównania. Gdy 1 euro = 4,20 zł, kod może przyjąć taką postać:

```
(49 * 4.2 >= 200);  // wyrażenia logiczne umieszcza się w nawiasie
```

Następnie skompiluj i uruchom program, aby sprawdzić poprawność wyniku.

```
1. #include <iostream>
2. using namespace std;
3.
4. int main() {
5.     cout << (49 * 4.2 >= 200) << endl;
6.     return 0;
7. }
```

ZMIENNE

Zmienne przechowują wartości określonego typu, m.in. liczby całkowite, liczby rzeczywiste, wartości logiczne i łańcuchy znaków. Najpierw trzeba zmienną **zadeklarować**, a potem przypisać jej wartość początkową. Dopiero wtedy można ją modyfikować. Każda zmienna w C++ powinna mieć krótką i znaczącą nazwę (bez polskich liter i spacji), np. `n` albo `liczba`. Należy przy tym pamiętać, że **wielkość liter jest istotna** – `N` i `n` to różne zmienne. Podczas deklaracji należy podać typ danych, jaki będzie przechowywany w zmiennej. Podstawowe typy danych to:

- `int` – liczba całkowita,
- `float` – liczba rzeczywista,
- `char` – pojedynczy znak,
- `bool` – wartość logiczna (`false`, `true`).

Aby obliczyć i wyświetlić sumę dwóch liczb, oprócz zmiennych przechowujących wartość składników należy zadeklarować również zmienną zawierającą wynik dodawania i wpisać jej nazwę w poleceniu `cout`.

```
1. #include <iostream>
2. using namespace std;
3.
4. int main() {
5.     int x = 34;
6.     int y = 66;
7.     int wynik = x + y;
8.     cout << wynik << endl;
9.     return 0;
10.}
```

Rys. 7. Obliczanie sumy zmiennych całkowitych `x` i `y`

Ćwiczenie 3. Średni dystans

W czasie wakacji uczniowie na obozie wędrownym przeszli w ciągu kolejnych trzech dni 12, 14 i 18 km. Oblicz średnią długość pokonywanych tras.

- Zadeklaruj zmienne i przypisz im podane wartości.

```
float a = 12;
float b = 14;
float c = 18;
```

- Zapisz odpowiedni wzór, za pomocą którego wyznaczysz średnią podanych wartości.

```
float srednia = (a + b + c) / 3;
```

- Skompiluj i uruchom program, aby sprawdzić poprawność wyniku. Jak myślisz, dlaczego użyto tutaj typu **float**, a nie **int**?

```
1. #include <iostream>
2. using namespace std;
3.
4. int main() {
5.     float a = 12;
6.     float b = 14;
7.     float c = 18;
8.     float srednia = (a + b + c) / 3;
9.     cout << srednia << endl;
10.    return 0;
11.}
```

DEFINIOWANIE FUNKCJI

Funkcje pozwalają podzielić większy problem na mniejsze, względnie niezależne części, a także przyśpieszają i ułatwiają pisanie programu, który staje się dzięki nim bardziej zwięzły i przejrzysty. Definicję umieszcza się powyżej funkcji **main**.Przed nazwą funkcji podaje się typ wyniku funkcji, następnie jej nazwę (bez polskich znaków, spacji) i wstawia nawiasy, w których można umieścić parametr lub parametry. Każdy parametr poprzedzony jest typem danych. Treść funkcji zawiera się w nawiasach klamrowych. Kolejne wiersze kodu zawierają instrukcje do wykonania. Polecenie **return**, które przekazuje szukaną wartość, kończy działanie funkcji.

ZAPAMIĘTAJ!

```
<typ wyniku> <nazwa funkcji>(<typ parametru> <parametr>) {
    <instrukcje>
}
```

Ćwiczenie 4. Przeliczanie prędkości

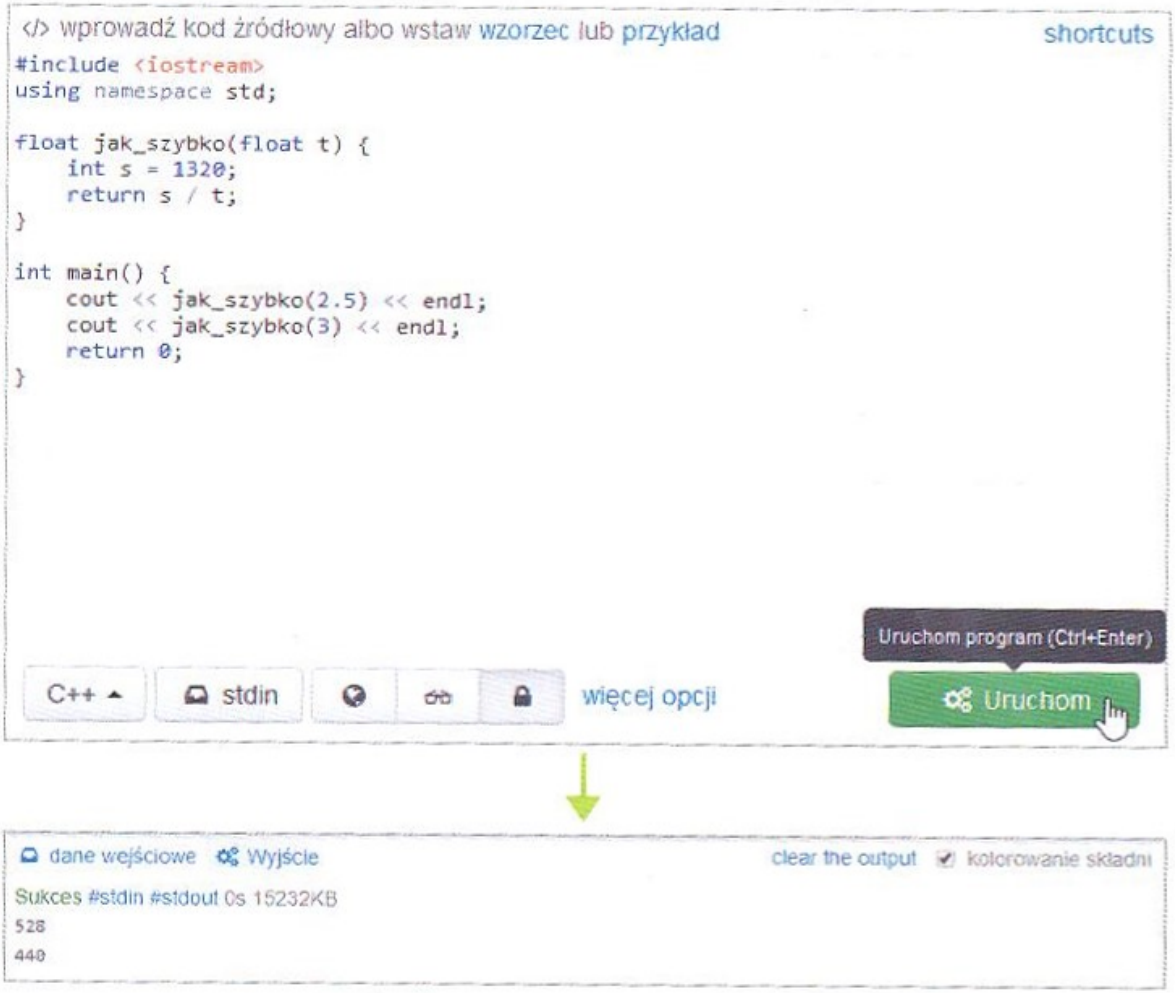
Samolot z Warszawy do Rzymu pokonuje dystans 1320 km. Przeanalizuj dane w tabeli i zdefiniuj funkcję **jak_szybko(float t)**, której parametrem jest czas przelotu samolotu w godzinach, a wynikiem średnia prędkość tego samolotu.

Wywołanie funkcji	Wynik
jak_szybko(2.5)	528.0
jak_szybko(3)	440.0

- Aby wyznaczyć średnią prędkość, należy podzielić drogę przez czas.
- Zdefiniuj funkcję **jak_szybko(float t)**, np.

```
1. float jak_szybko(float t) {
2.     int s = 1320;
3.     return s / t;
4. }
```

- Sprawdź działanie programu. Podczas jednego uruchomienia funkcji **main()** można sprawdzić wywołanie funkcji z różnymi danymi. Wystarczy wstawić kilka wywołań danej funkcji, tak jak pokazano to na zrzucie poniżej.



Ćwiczenie 5. Przeliczanie jednostek

Mila lądowa, jednostka długości stosowana w krajach anglosaskich, równa jest 1,609 km. Przeanalizuj dane w tabeli i zdefiniuj funkcję **zamiana(float m)**, której parametrem jest odległość w milach lądowych, a wynikiem ta sama odległość w kilometrach.

Wywołanie funkcji	Wynik
zamiana(10)	**16.09**
zamiana(12)	**19.308**

- Aby zamienić mile na kilometry, należy pomnożyć odległość w milach przez 1,609.
- Zdefiniuj funkcję **zamiana(float m)** i sprawdź jej działanie z różnymi danymi.

```
1. #include <iostream>
2. using namespace std;
3.
4. float zamiana(float m) {
5.     return m * 1.609;
6. }
7.
8. int main() {
9.     cout << zamiana(10) << endl;
10.    cout << zamiana(12) << endl;
11.    return 0;
12.}
```

→

```
16.09
19.308
```

Ćwiczenie 6. O której godzinie pociąg dotrze do celu?

Nocny pociąg z Krakowa do Paryża wyjeżdża ze stacji o godzinie 21. Przeanalizuj dane w tabeli i zdefiniuj funkcję **godzina(int n)**, której parametrem jest czas przejazdu pociągu w godzinach, a wynikiem godzina dojazdu do celu.

Wywołanie funkcji	Wynik
godzina(14)	**11**
godzina(20)	**17**

- 21 + 14 = 35, 21 + 20 = 41. Doba trwa 24 godziny, dlatego aby obliczyć czas przyjazdu pociągu, należy wyznaczyć resztę z dzielenia wyniku dodawania przez 24.
- Zdefiniuj funkcję **godzina(int n)** – wykorzystaj do obliczeń resztę z dzielenia całkowitego.

```
1. int godzina(int n) {
2.     return (21 + n) % 24;
3. }
```

- Sprawdź działanie funkcji z różnymi danymi. Jakich danych warto użyć?

Ćwiczenie 7. Która godzina będzie w Nowym Jorku?

Sprawdź, jaka strefa czasowa obowiązuje w Nowym Jorku, a następnie przeanalizuj dane w tabeli i zdefiniuj funkcję **`godz_nj(int g)`**, której parametrem jest pełna godzina w Warszawie, a wynikiem aktualna pełna godzina w Nowym Jorku.

Wywołanie funkcji	Wynik
`godz_nj(18)`	`12`
`godz_nj(2)`	`20`

ZAPAMIĘTAJ!

Ważną umiejętnością programistyczną jest zapis w języku formalnym tego, co potrafimy wyliczyć bez komputera. Podczas rozwiązywania zadań staramy się najpierw zrozumieć problem, przeanalizować go na kilku przykładach, potem zaproponować algorytm i zapisać go w języku programowania. Nie wolno zapomnieć o testowaniu. Wypracowanie takiej metody postępowania ułatwia rozwiązywanie różnych problemów, nie tylko wywodzących się z informatyki.

PODSUMOWANIE

- C++ to język kompilowany, za pomocą którego można pisać mniej i bardziej skomplikowane programy. Napisane przez człowieka programy są zamieniane na język maszynowy, dopiero potem mogą być wykonywane przez komputer.
- Bloki oznacza się nawiasami klamrowymi, dla większej przejrzystości warto stosować również wcięcia.
- Język C++ rozróżnia małe i wielkie litery.
- Rozwiązywany problem warto podzielić na mniejsze problemy, które można niezależnie rozwiązać przez definiowanie funkcji. Gdy problem jest prosty, wystarczy jedna funkcja.
- Główna funkcja w języku C++, w której wywołuje się inne funkcje, nosi nazwę `main()`.

- Funkcje mogą przekazywać wynik (polecenie **return**) lub powodować określony skutek na ekranie (instrukcja **cout**).
- Podstawowe operatory arytmetyczne i porównania są zbliżone do używanych w języku naturalnym, dlatego po napisaniu kilku programów korzystanie z nich staje się intuicyjne.

PYTANIA SPRAWDZAJĄCE

1. Jaka liczba będzie wynikiem operacji `5 % 2`?
2. Czym się różni polecenie `cout << "a";` od polecenia `cout << a;`?
3. Do czego służy polecenie **return** w funkcji?

ZADANIA DODATKOWE

Zadanie 1. Ile kosztuje?

Zdefiniuj funkcję **cena(float f)**, której parametrem jest cena w dolarach funta danego produktu, a wynikiem cena w złotówkach kilograma tego produktu. Sprawdź w internecie aktualny kurs dolara oraz przelicznik jednostek masy (funt i kilogram).

Zadanie 2. Prędkość w skali Beauforta

Zdefiniuj funkcję **skala(float v)**, której parametrem jest prędkość wiatru w węzłach, a wynikiem przeliczona prędkość w skali Beauforta.

Zadanie 3. Która godzina będzie w Warszawie?

Różnica czasu między Los Angeles a Warszawą wynosi dziewięć godzin – jeżeli w LA zegar wskazuje godzinę 21, to w Warszawie jest godzina 6 rano. Przeanalizuj dane w tabeli i zdefiniuj funkcję **godz_waw(int g)**, której parametrem jest pełna godzina w Los Angeles, a wynikiem – aktualna godzina w Warszawie.

Wywołanie funkcji	Wynik
`godz_waw(8)`	17
`godz_waw(21)`	6