

## **Įvadas ..... 4**

### **1 SKYRIUS**

#### **Saugus darbas kompiuteriu ..... 6**

**TEMA +** Saugus ir sveikatą tausojantis darbas skaitmeniniu įrenginiu ..... 8

**TEMA +** Saugaus elgesio virtualiojoje erdvėje principai, pavojai ir sunkumai ..... 10

### **2 SKYRIUS**

#### **Algoritmai ir programavimas..... 12**

2.1. Duomenų rikiavimo ir paieškos algoritmai (skaičiuoklė) ..... 14

**TEMA +** Duomenų kodavimas ir skaičiavimo sistemos kompiuteriuose ..... 22

**TEMA +** Kompiuterių raida, algoritmai ir programos ..... 30

**TEMA +** Programavimo kalbos konstrukcijos. Programavimo aplinka ..... 38

**TEMA +** Algoritmų parinkimas: sveikieji ir realieji skaičiai ..... 45

2.2. Algoritmų parinkimas: šakojimo sakiny...50

2.3. Programos derinimas.  
Programos rezultatų pateikimas ..... 57

2.4. Išorinių duomenų naudojimas ..... 60

2.5. Išorinių duomenų naudojimas.  
Algoritmų parinkimas: cikliniai algoritmai.....64

2.6. Problemų sprendimo automatizavimas.....71

### **3 SKYRIUS**

#### **Aparatinė ir programinė įranga ..... 78**

3.1. Kompiuterio pagrindinių struktūrinių dalių paskirtis ir funkcijos ..... 80

3.2. Programinė įranga. Operacinė sistema .... 83

### **4 SKYRIUS**

#### **Elektroninės paslaugos ..... 88**

## ĮVADAS

Gali būti, kad mintis mokytis iš „senamadiško“ popierinio vadovėlio, kaip programuoti, elgtis atsakingai internete ar taisyklingai sėdėti prie kompiuterio, tau atrodys mažų mažiausiai keista. Taip ir girdžiu sakant: „Juk galima paieškoti gūgle ar jutube – yra daugybė pamokų! O dar geriau paklausti dirbtinio intelekto!“ Tačiau kai kruopščiai kursai pirmąsias kodo eilutes, fizinis puslapis, kuriame gali piršto galiuką tvirtai prispausti prie tolesnio žingsnio aprašymo, užtikrintai ves tave ir neleis pasimesti.

Netrukus sužinosi, kad šis vadovėlis gali tapti tikru tavo pagalbininku per informatikos pamokas ir palydovu į paslaptinę programavimo pasaulį. Sveikinu tave, atradus gyvenimą už jutubo ribų!

Technologijos nebėra nišinė tema. Iš tikrųjų šiandien kiekviena pramonės šaka remiasi technologijomis, todėl pasirinkimas paprastas: arba būti tame pasaulyje ir tobulėti kartu su juo, arba likti nuošalyje.

Senovės graikų filosofas Heraklitas yra pasakęs, kad „vienintelis pastovus dalykas gyvenime yra pokytis“. Ir tai ypač tinka kompiuterijos pasauliui.

Ar yra „aitišnikų“ žargonas? Be abejo. Ir žargonas, ir susitarimai, ir juokeliai, kuriuos supras tik programuotojai. Ar galima gyventi be pagrindinių programavimo žinių? Taip. Ar gebėjimas programuoti padės tau suprasti pasaulį, kuriame dabar gyvename, ir geriau jame jaustis? Manau, taip. Ar tai gali pastūmėti tave nuostabios profesijos link? Neabejotinai, jei to nori. Vis dar beviltiškai trūksta technologijų specialistų. Galimybių yra, bet jų neatrasi, jei bent šiek tiek neperprasi programavimo kalbos.

Tie, kurie domisi programavimu ar susimąsto apie pavojus, kokie gali būti virtualiojoje erdvėje, ir būdus, kaip būtų galima jų išvengti, šiame vadovėlyje ras daug vertingos, išties sudėtingos, bet paprastai išdėstytos informacijos, kurią surinko ir apibendrino autorė – patyrusi informatikos, matematikos, STEAM mokytoja.





# 1.

## SAUGUS DARBAS KOMPIUTERIU

Gyvename skaitmeniniame amžiuje, kai internetas ir informacinės technologijos (toliau – IT) tampa vis svarbesni žmogaus gyvenime. Išmanieji telefonai ir planšetiniai kompiuteriai užima daug mūsų laiko. Jie atveria naujų galimybių, bet kartu ir keičia žmonių bendravimo būdą. Todėl būtina išsiugdyti sveikus šių įrenginių naudojimo įpročius, kaip ir žinoti apie galimus pavojus virtualiojoje erdvėje, kad sugebėtum save apsaugoti.

Šiame skyriuje mokysies:

- kokias grėsmes gali kelti IT;
- kokie pavojai tyko internete;
- kaip atpažinti elektronines patyčias;
- kas yra kibernetinis saugumas ir kaip apsaugoti savo asmens duomenis.





Ilgai dirbant kompiuteriu ar kitu IT įrenginiu gali atsirasti įvairių fizinės ir psichinės sveikatos problemų.

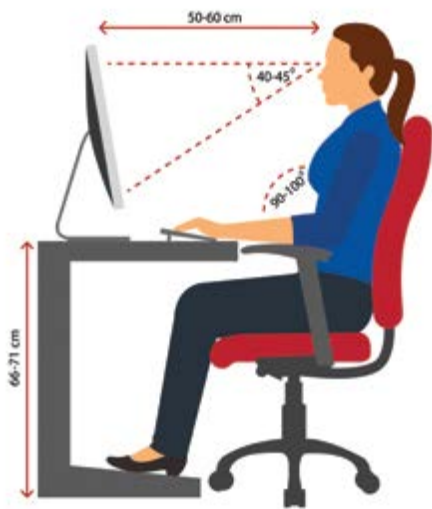
**Grėsmės fizinei gerovei:**

**Raumenų ir kaulų sistemos sutrikimai.** Ilgalais sėdėjimas prie kompiuterio ar telefono gali sukelti kaklo, pečių ir nugaros skausmus, riešo kanalo sindromą, pabloginti laikyseną.

**Regėjimas.** Dėl ilgo žiūrėjimo į ekraną gali atsirasti akių nuovargis, sausumas, dirglumas ir netgi regėjimo ydos.

**Miego sutrikimai.** Mėlyna šviesa, kurią spinduliuoja ekranai, gali sutrikdyti melatonino gamybą ir lemti miego sutrikimus.

**Nutukimas.** Jei daug laiko praleidžiama prie skaitmeninio įrenginio, didėja nutukimo rizika, nes mažėja fizinis aktyvumas.



**Akių nuovargis**

Ilgai žiūrėdamos į ekraną, akys įsitempia, o tai gali sukelti sausumo, deginimo ir dirginimo pojūtį. Taip pat gali išsivystyti regėjimo ydos, tokios kaip trumparegystė.



**Riešo kanalo sindromas**



Dėl suspaustų nervų riešo kanale gali skaudėti ranką, pradėti dilgčioti, tirpti rankos ir pirštai.

**Nugaros skausmas**

Jei ilgai ir netaisyklingai sėdime, įsitempia nugaros raumenys ir gali pradėti skaudėti nugarą.



**Galvos skausmas**



Nuolatinė akių ir kaklo raumenų įtampa gali sukelti galvos skausmus.

**Darbo aplinkos higiena**

- Natūrali šviesa turi kristi iš šono.
- Dirbtinė šviesa turi apšviesti dokumentą, ne ekraną.
- Veikiantis kompiuteris didina kambario temperatūrą ir sausina orą, todėl patalpa turi būti vėdinama, kartais ir drėkinama.

**Grėsmės psichinei gerovei:**

**Priklausomybė.** Pernelyg dažnas ir ilgas skaitmeninių įrenginių naudojimas gali peraugti į priklausomybę, kuri paveikia socialinius santykius, darbą ar mokslus.

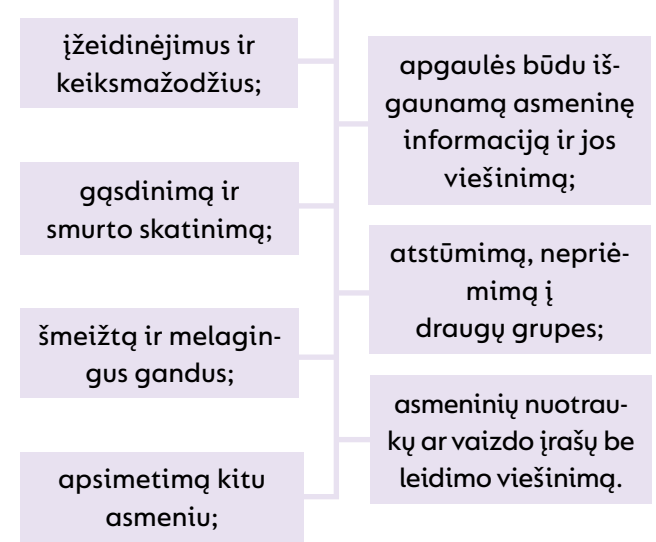
**Stresas ir nerimas.** Nuolatinis informacijos srautas ir socialinės žiniasklaidos pranešimai gali sukelti stresą ir nerimą.

**Depresija.** Pernelyg ilgas naršymas internete gali padidinti depresijos riziką.

**Patyčios internete.** Patyčios internete gali sukelti emocinę traumą ir paveikti psichinę sveikatą.

**Patyčios internete** – tai bet koks tyčinis ir agresyvus elgesys, nukreiptas į kitą asmenį internete ar kitoje virtualiojoje erdvėje.

Jos apima:



**Kaip išvengti grėsmių ar jas sumažinti:**

**Raumenų ir kaulų sistemos sutrikimai.** Reguliariai daryti trumpų mankštų, pasivaikščiojimo pertraukas, tinkamai susireguliuoti darbo vietą. Laisvalaikiu sportuoti.

**Regėjimas.** Leisti pailsėti akims, daryti pertraukas nuo ekrano, akių mankštą ir reguliariai tikrintis akis.

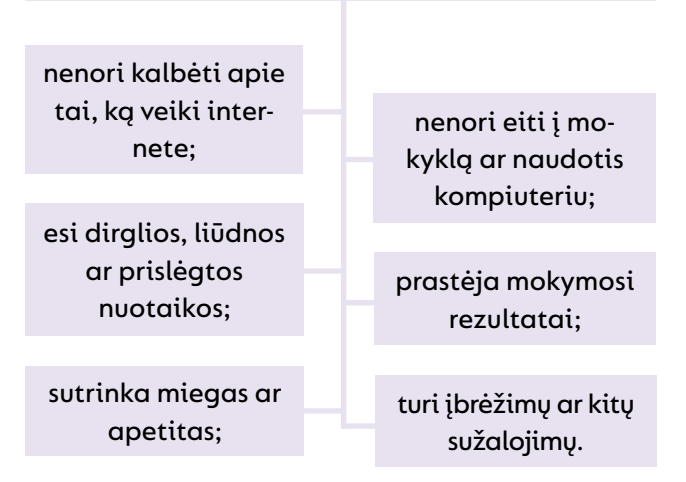
**Miego sutrikimai.** Prieš miegą vengti ekranų, miegoti tamsiame kambaryje ir laikytis tvarkingo miego režimo.

**Nutukimas.** Reguliariai sportuoti ir išsiugdyti įprotį daryti trumpas mankštos pertraukėles.

**Psichinė sveikata.** Riboti laiką internete, bendrauti su artimaisiais ir prareikus būtinai kreiptis pagalbos į specialistus.

**Kaip atpažinti, kad patiri patyčias internete?**

Yra keletas ženklų, kurie gali rodyti, kad patiri patyčias internete:



- KLAUSIMAI**
- 1 Aptarkite sveikatos problemas, kurias dažniausiai sukelia darbas kompiuteriu.
  - 2 Kokios yra akių nuovargio pasekmės? Kaip būtų galima jų išvengti?

**Sąlygos sakiny IF „Python“ programavimo kalboje**

„Python“ programavimo kalboje sąlygos sakiny IF vykdo programą, atsižvelgdamas į duotos sąlygos teisingumą arba klaidingumą. Šio sakiny sintaksė:

**IF sąlyga:**  
Sakiniai, kai sąlyga tenkinama.

**ELSE:**  
Sakiniai, kai sąlyga netenkinama.

**Sakiniai, kai sąlyga tenkinama**

Kodo blokas vykdomas, jei sąlyga yra teisinga. Šis blokas turi būti atitrauktas nuo kairiojo krašto.

**Sakiniai, kai sąlyga netenkinama**

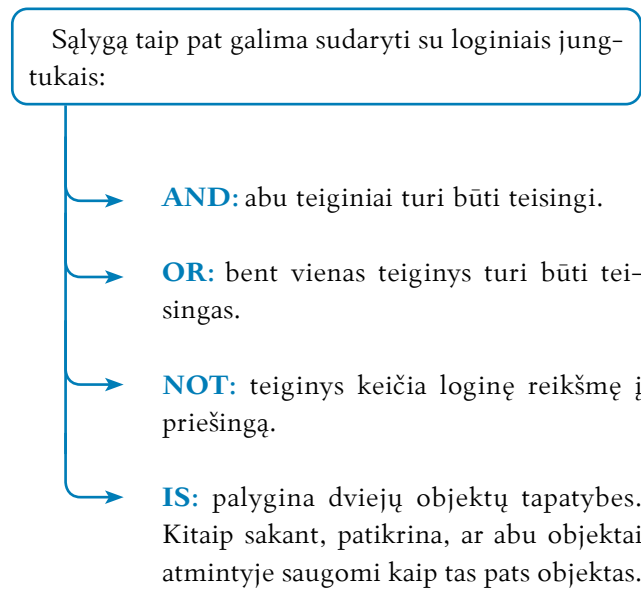
Galima rinktis sakiny ELSE. Jis apibrėžia kodą, kuris bus vykdomas, jei sąlyga yra klaidinga. Šio sakiny kodo blokas taip pat turi būti atitrauktas.

**Loginės sąlygos**

Loginės sąlygos apibrėžiamos taip:

==	Tikrinama, ar kintamieji lygūs.
!=	Tikrinama, ar kintamieji nelygūs.
<	Tikrinama, ar pirmas kintamasis mažesnis už antrą.
>	Tikrinama, ar pirmas kintamasis didesnis už antrą.
<=	Tikrinama, ar pirmas kintamasis mažesnis už antrą arba lygus jam.
>=	Tikrinama, ar pirmas kintamasis didesnis už antrą arba lygus jam.
	Tikrinama, ar bent vienas iš kintamųjų tinka pagal sąlygą (loginis jungtukas <i>Arba</i> ).
&&	Tikrinama, ar visi kintamieji tinka pagal sąlygą (loginis jungtukas <i>Ir</i> ).

**Loginiai jungtukai**

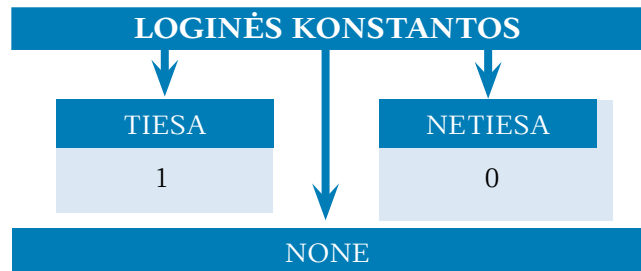


**Skirtumas tarp YRA ir ==**

Operacijos ženklas == lygina objektų reikšmes. Jei du kintamieji turi tą pačią teksto ar skaičiaus reikšmę, == rodys *Tiesa*.

IS lygina objektų tapatybę. Net jei du kintamieji turi tą pačią teksto ar skaičiaus reikšmę, IS rodys *Tiesa* tik tada, kai abu kintamieji atmintyje bus saugomi kaip tas pats objektas.

```
x = 10
y = 10
print(x == y) #True, nes reikšmės yra lygios
print(x is y) #False, nes objektai atmintyje yra skirtingi
```



Speciali „Python“ programavimo kalbos konstanta, kuri nurodo neapibrėžtą reikšmę, arba tuščią objektą. Ji nėra tapati 0, *Netiesa* arba tuščiai eilutės (*string*) tipo kintamojo reikšmei.

**Sąlygos sakiny ELIF „Python“ programavimo kalboje**

„Python“ programavimo kalboje sakiny ELIF (trumpinys iš ELSE IF) patikrina kelias skirtingas sąlygas viename sąlyginio sakiny bloke. Tai yra sąlygos sakiny IF alternatyva, ji praverčia, kai reikia patikrinti daugiau nei vieną sąlygą. Sąlygos sakiny ELIF sintaksė:

**IF sąlyga1:**  
# Kodas bus vykdomas, jei sąlyga1 yra teisinga.

**ELIF sąlyga2:**  
# Kodas bus vykdomas, jei sąlyga1 yra klaidinga, o sąlyga2 – teisinga.

**ELIF sąlygaN:**  
# Kodas bus vykdomas, jei sąlyga1 ir sąlyga2 yra klaidingos, o sąlygaN – teisinga.

**ELSE:**  
# Kodas bus vykdomas, jei visos sąlygos klaidingos.

```
if skaicius > 15:
    print("Skaicius yra didesnis nei 15")
elif skaicius > 10:
    print("Skaicius yra didesnis nei 10, bet mazesnis nei 15")
else:
    print("Skaicius yra mazesnis arba lygus 10")
```

- Sakiniai ELIF po sakiny IF gali kartotis nenustatytą kiekį kartų.
- Sąlygos tikrinamos iš eilės. Jei pirma sąlyga teisinga, likusios sąlygos nėra tikrinamos.
- Sakiny ELSE nebūtinai. Jei jo nėra, o visos sąlygos yra klaidingos, programa bus nevykdoma.

**TEORIJA PAVYZDŽIAIS**

**1** Programa, tikrinanti, ar įvestas sveikasis skaičius *x* yra lyginis.

```
x = int(input("Iveskite sveika skaiciu: "))
#Patikrinama, ar skaicius yra lyginis
if x % 2 == 0:
    print("Skaicius ", x, "yra lyginis")
else:
    print("Skaicius ", x, "yra nelyginis")
```

**2** Programa, kuri įvestą skaičių nuo 1 iki 5 užrašo žodžiu.

```
skaicius = int(input("Iveskite skaiciu nuo 1 iki 5: "))
#Skaiciaus konvertavimas
if skaicius == 1:
    print("Vienas")
elif skaicius == 2:
    print("Du")
elif skaicius == 3:
    print("Trys")
elif skaicius == 4:
    print("Keturi")
elif skaicius == 5:
    print("Penki")
else:
    print("Ivestas skaicius yra ne is intervalo [1;5]")
```

**UŽDUOTYS**

- Kad paskatintų nuosekliai dirbti, matematikos mokytoja nusprendė pusmečio gale parašyti po dešimtuką mokiniams:
  - kurie sąžiningai sprendė namų darbus ( $n1 = 1$ , kai mokinys sąžiningai sprendė namų darbus, ir  $n1 = 0$ , kai mokinys atliko ne visus namų darbus arba sprendė juos nesąžiningai);
  - kurių užrašai tvarkingi ir išsamūs ( $n2 = 1$ , kai mokinio užrašai tvarkingi ir išsamūs,  $n2 = 0$ , kai užrašai netvarkingi arba juose trūksta dalies informacijos);
  - kurie be pateisinamos priežasties nepraleido nė vienos pamokos ( $n3 = 1$ , kai mokinys be pateisinamos priežasties nepraleido nė vienos pamokos,  $n3 = 0$ , kai mokinys neatvyko į pamokas be pateisinamos priežasties).

**Testavimas:**

Įvedami duomenys			Rezultatas
n1	n2	n3	
0	0	0	Mokinys dešimtuko negaus.
0	1	1	Mokinys dešimtuko negaus.
1	1	1	Mokinys dešimtuką gaus.

**C++ ir „Python“ programavimo kalbomis parašyk programą, kuri praneštų, ar mokinys gaus dešimtuką už gerą pusmečio darbą.**

- C++ ir „Python“ programavimo kalbomis parašyk programą, kuri praneštų, kada baigiasi pamoka, jei iki pamokos pabaigos liko t minučių:
  - daugiau negu 30 min., turi būti spausdinamas pranešimas „Liko dar labai daug laiko“;
  - mažiau negu 30 min., bet daugiau negu 15 min., turi būti spausdinamas pranešimas „Liko dar nemažai laiko“;
  - 7-15 min. – „Liko nedaug laiko“;

- 1-6 min. – „Pamoka beveik baigiasi“;
- 0 min. – „Pamoka baigėsi“;
- mažiau negu 0 arba ne skaičius, turi atsiųsti pranešimas „Blogai įvesti duomenys“.

**Testavimas:**

Įvedami duomenys	Rezultatas
Kiek liko iki pamokos pabaigos? 40	Liko dar labai daug laiko.
Kiek liko iki pamokos pabaigos? 5	Pamoka beveik baigiasi.
Kiek liko iki pamokos pabaigos? -3	Blogai įvesti duomenys.

- Jonas žino, kad mokytoja jam duos spręsti vieną iš trijų kontrolinio darbo variantų. Atlikdamas užduotį su sveikaisiais skaičiais a ir b, jis turės apskaičiuoti x reikšmę:

Pirmas variantas:  $x = \sqrt{a + b} + 3ab$

Antras variantas:  $x = \frac{a}{b} - 2b$

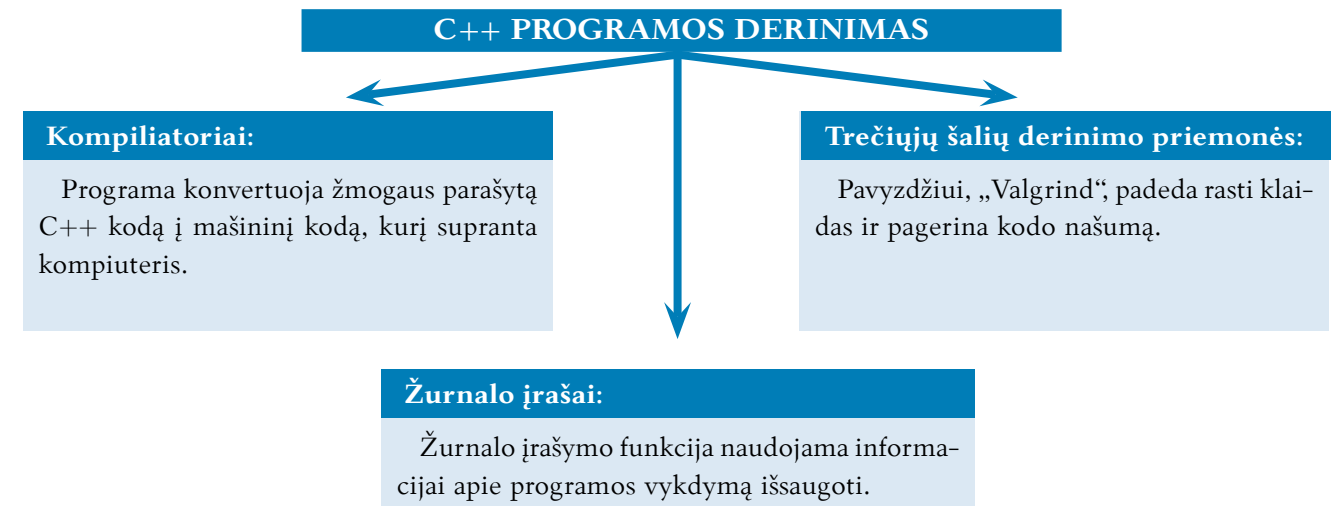
Trečias variantas:  $x = |a| + b^a$

**C++ ir „Python“ programavimo kalbomis parašyk programą, kuri padėtų Jonui patikrinti, ar jis teisingai apskaičiavo x reikšmę sprenddamas savo varianto užduotį.**

**Testavimas:**

Įvedami duomenys	Rezultatas
a = 3 b = 7 Koks variantas? 1	x = 66,16228
a = 0 b = 0 Koks variantas? 2	Negalima b reikšmė.
a = 2 b = 2 Koks variantas? 3	x = 6.
a = -3 b = 2 Koks variantas? 1	Negalima a ir (arba) b reikšmė, a ir b suma turi būti >=0.

**Programos klaidų paieškos ir taisymo procesas vadinamas derinimu. Klaidos gali būti įvairios – nuo paprastų rašybos iki kodo logikos riktų. Ir C++, ir „Python“ turi įvairių priemonių, kurios padeda ištaisyti klaidas.**



**C++ kompiliatoriai**

Kiekvienas kompiliatorius turi savo privalumų ir trūkumų. Keli populiarūs C++ kompiliatoriai:

- „GNU Compiler Collection“ (GCC): labiausiai paplitęs, galingas ir universalus atvirojo kodo kompiliatorius. Nemokamas. Jį palaiko daugelis platformų, tarp jų – „Linux“, „Windows“ ir „macOS“. Kompiliatorių rasi <https://gcc.gnu.org>.
- „Clang“: modernus ir greitas atvirojo kodo kompiliatorius. Integruotas su daugeliu IDE. Geras pasirinkimas pradedantiesiems. Kompiliatorių rasi <https://clang.llvm.org>.
- „Visual Studio“: komercinė „Microsoft“ IDE, kurioje yra integruotas C++ kompiliatorius. Turi daug priemonių ir funkcijų, palengvinančių C++ programų kūrimą ir derinimą. Tinka profesionaliems programuotojams.
- „Intel C++ Compiler“ (ICC): komercinis „Intel“ kompiliatorius, garsėjantis našumu. Gali pagerinti programų vykdymo greitį.

