

# Užduočių Gidas (Trumpa versija)

**Tikslas:** Greitas start - tik essentials!

**Pilna versija:** [UzduotiuGidas-full.md](#) ← Detalesnė info

## 🔗 Quick Start (5 žingsniai)

### 1. GitLab projektas

```
# GitLab → New Project → cpp-2026
# Settings → Members → Add dėstytoja (Reporter)
git clone https://gitlab.mif.vu.lt/[username]/cpp-2026.git
cd cpp-2026
```

### 2. Projekto struktūra

```
cpp-2026/
├── README.md           ← Naudok šabloną: README-project.md
└── .gitignore
└── U1/
    ├── README.md       ← Naudok šabloną: README-assignment.md
    ├── 01/              ← Žingsnis 1
    ├── 02/              ← Žingsnis 2
    └── 03/              ← Žingsnis 3 (FINAL)
```

**README šablonai:** [README-templates/](#) direktoriuje

### 3. Git workflow (po kiekvieno žingsnio!)

```
# Po žingsnio 1
git add U1/01/
git commit -m "U1: 1 žingsnis - [aprašymas]"
git push

# Po žingsnio 2
git add U1/02/
git commit -m "U1: 2 žingsnis - [aprašymas]"
git push

# ... ir t.t.
```

**SVARBU:** Commit'inti **po kiekvieno žingsnio**, ne visą užduotį vienu kartu!

## 4. README.md užpildymas

**Po visų žingsnių:**

```
cd U1
# Nukopijuok šabloną
cp ../README-templates/README-assignment.md README.md
# Redaguok (užpildyk lenteles, testus)
git add README.md
git commit -m "U1: README"
git push
```

## 5. Moodle pateikimas

```
# Sukurti archyvą
cd cpp-2026
git archive --format=zip --output=U1_VardasPavarde.zip HEAD U1/ README.md
.gitignore

# Įkelti į Moodle + pridėti GitLab URL
```

## README hierarchija

Lygis	Failas	Šablonas	Privaloma?
Projektas	/README.md	README-project.md	<input checked="" type="checkbox"/> TAIP
Užduotis	/U1/README.md	README-assignment.md	<input checked="" type="checkbox"/> TAIP
Žingsnis	/U1/01/README.md	README-step.md	<input checked="" type="checkbox"/> NE

## Vertinimas

Kriterijus	%
Funktionalumas	50%
Kodo kokybė	30%
Git + README	20%

**Commit'ų vertinimas:**

- Po kiekvieno žingsnio: 100%

- Keli commit'ai: 70%
  - Vienas commit: 30%
- 

## ⚙️ .gitignore (būtinės!)

Sukurti `.gitignore` root direktoriuje:

```
# Compiled files
*.o
*.out
programa
a.out

# Editor files
*~
.vscode/
.idea/
*.swp

# OS files
.DS_Store
```

---

## ❓ DUK (Top 5)

### K: Ar galiau naudoti branch'us?

A: Taip, bet **subdirektorijos paprastesnės**. Branch'ai - optional.

### K: Makefile būtinės?

A: Nuo 3 žingsnio - **rekomenduojamas**. Anksčiau - `g++` tiesiogiai OK.

### K: Ar reikia logging destruktoriuose?

A: **Taip** - padeda debug'inti (U2+).

### K: Self-assignment check?

A: **Būtinės** operator= (U4).

### K: Išimtys - kada naudoti?

A: Pradedant U5 (validacija), U6+ (runtime errors).

**Daugiau:** [UzduotiuGidas-full.md](#)

---

## ✉️ Nuorodos

- **README šablonai:** [README-templates/](#)
- **Pilnas gidas:** [UzduotiuGidas-full.md](#)
- **Užduotys:** [U1.md](#), [U2.md](#), ...
- **GitLab Help:** <https://docs.gitlab.com/>

- **C++ Reference:** <https://en.cppreference.com/>
- 

**Kontakta:**

Dėstytojas: [email]

Konsultacijos: [laikas/vieta]

---

Štai ir viskas! Sėkmės! 