

Programování v C

Jan Faigl

Katedra počítačů
Fakulta elektrotechnická
České vysoké učení technické v Praze

Organizace předmětu
BAB36PRGA – Programování v C

Přehled témat

- Část 1 – Organizace předmětu
 - Organizace
 - Cíle
 - Prostředky dosažení cílů
 - Hodnocení a zkouška
 - Komunikace
 - Vývojová prostředí a služby akademické sítě

Část I Organizace předmětu

Předmět a přednášející

BAB36PRGA – Programování v C

- Webové stránky předmětu
<https://cw.fel.cvut.cz/wiki/courses/bab36prga>
- Odevzdávání domácích úkolů
<https://cw.felk.cvut.cz/brute>
- Přednášející:
 - prof. Ing. **Jan Faigl**, Ph.D.
 - Katedra počítačů – <https://cs.fel.cvut.cz>
 - Centrum umělé inteligence – **Artificial Intelligence Center (AIC)**
<https://aic.fel.cvut.cz>
 - Centrum robotiky a autonomních systémů
Center for Robotics and Autonomous Systems – CRAS
<https://robotics.fel.cvut.cz>
 - **Laboratoř výpočetní robotiky (Computational Robotics Laboratory)**
<https://comrob.fel.cvut.cz>



<https://aic.fel.cvut.cz>

<https://robotics.fel.cvut.cz>

<https://comrob.fel.cvut.cz>

Cvičení

- Bc. Jan Feber
- Bc. David Valouch






Přednášky – letní semestr (ZS) akademického roku 2023/2024

- Harmonogram akademického roku 2023/2024.
<https://www.fel.cvut.cz/cz/education/harmonogram2324.html>
- Přednášky
 - Kampus Dejvice, místnost T2:D3-209, úterý, 18:00–19:30.
- 14 výukových týdnů. *13+1 přednášek*
- Státní svátky:
 - 29.03.2024 (pátek), 01.04.2024 (pondělí), 01.05.2024 (středa), 08.05.2024 (středa),
- Rektorský den (den bez výuky) – 14.05.2024 (úterý).
- Náhrada výuky 09.05.2024 (čtvrtek) bude středeční výuka za 08.05.2024 (středa).
- *Předpokládaný test z fyziky (B3B02FY1A) v týdnu od 8.4. a v týdnu od 6.5.*
- Konzultace: po domluvě. *Typicky pátek odpoledne na KN, KN:E-205.*



Organizace a hodnocení BAB36PRGA – Programování v C

- Rozsah: 2p+2c; Zkončení: Z,ZK; Kredity: 6; 1 ECTS kredit je 25–30 hodin za semestr, cca 180 h.
 - Kontaktní část (přednáška a cvičení): 3 hodiny týdně, tj. 42 hodin celkem.
 - Zkouška včetně přípravy: 10 hodin.
 - Domácí příprava (úkoly) cca **9 hodin týdně.** *Medián zátěže!*
- **Průběžná práce v semestru** – domácí úkoly.
- **Zkouškový test a implementační zkouška.** *Schopnost samostatné práce na počítačích v učebnách.*
- Docházka na **cvičení** a odevzdání domácích úloh. *Samostatná práce (kontrola plagiatů).*
 - Postupujte systematicky, budete tak postupně rozvíjet své schopnosti.
 - Využijte čas v prvních úlohách a naučte se psát programy správně.
Program musí být nejen správný a funkční, ale také čitelný a udržovatelný!
- **Konzultace** – Pokud nevíte, tápete nebo řešíte domácí úkol příliš dlouho, **konzultujte** s cvičícím/přednášejícím. *Čtěte (učebnici), pochopte principy (nejen hledat řešení), hlídejte si čas a včas konzultujte!*
 - **Maximálně využijte kontaktní čas na cvičení/přednášce, ptejte se, diskutujte!**



Zdroje a literatura

- **Knihy (učebnice)** *Základní učební text „Programming in C“ (Kochan, 2014)*
 -  Programming in C, 4th Edition, Stephen G. Kochan, Addison-Wesley, 2014. *Recommended textbook.*
 -  C Programming: A Modern Approach, 2nd Edition, K. N. King, W. W. Norton & Company, 2008. *More like a reference manual, still comprehensive textbook.*
 -  The C Programming Language, 2nd Edition (ANSI C), Brian W. Kernighan, Dennis M. Ritchie, Prentice Hall, 1988 *1st edition 1978*
- Přednášky – podpora učebního textu, slidy, videa, poznámky a **vlastní poznámky.** *Součástí přednášek jsou také zdrojové kódy s příklady!*
- Cvičení – získání praktických dovedností řešením domácích úkolů a dalších úloh. *programovat, programovat, programovat*



Další učebnice jazyka C

 **Practical C Programming**, Steve Oualline, O'Reilly Media, Inc., 3rd edition, 1997) 



Briefer than Kochan's textbook, still comprehensive.

 **Effective C: An Introduction to Professional C Programming**, Robert C. Seacord, William Pollock, 2020. 

Great if you already known some of C syntax and like to improve your skill further.

 **Fluent C, Principles, Practices, and Patterns**, Christopher Preschern, O'Reilly Media, Inc., 2022. 



Suitable if you like to know more about coding practices.

 **21st Century C: C Tips from the New School**, Ben Klemens, O'Reilly Media, 2012. 


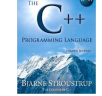
Cíle předmětu

- **Osvojit si** pohled na výpočetní prostředky jako „počítačový vědec“ a naučit se je efektivně používat. *Computer scientist*
 - Formulovat problém a jeho řešení počítačovým programem.
 - Získat povědomí jaké problémy lze výpočetně řešit.
- **Získat zkušenost** s programováním *ziskání vlastní zkušenosti*
 - Programování v C *cvičení, domácí úkoly, zkouška*
- **Osvojit si** schopnost číst, psát a porozumět malým programům
- **Získat** programovací návyky jak psát
 - Srozumitelné a přehledné zdrojové kódy;
 - Opakovaně použitelné programy.

Další zdroje

 **Introduction to Algorithms, 3rd Edition**, Cormen, Leiserson, Rivest, and Stein, The MIT Press, 2009, ISBN 978-0262033848 

 **Algorithms, 4th Edition**, Robert Sedgewick, Kevin Wayne, Addison-Wesley, 2011, ISBN 978-0321573513 

 **The C++ Programming Language, 4th Edition (C++11)**, Bjarne Stroustrup, Addison-Wesley, 2013, ISBN 978-0321563842 

Výuka programování

„Separating Programming Sheep from Non-Programming Goats“
<http://blog.codinghorror.com/separating-programming-sheep-from-non-programming-goats>
<http://www.eis.mdx.ac.uk/research/PhDArea/saeed/paper1.pdf>

- Efektivní metody výuky programování se hledají již od dob prvních počítačů *Déle než 50 let*
- Přesto se zdá, že je každý základní kurz programování obtížný a 30 % až 60 % studentů jej na poprvé nezvládnou.

Průchodnost PRGA je vyšší.
 2022/2023: 55 % (96 % z udělených zápočtů, 32)
 2021/2022: 46 % (89 % z udělených zápočtů, 38)
 2020/2021: 37 % (92 % z udělených zápočtů, 39)
 2019/2020: 65 % (100 % z udělených zápočtů, 23)

- **Základní koncept je pochopení principu přiřazení hodnoty proměnné!**

Je to především práce s pamětí, která je v Cčku velmi přímá.

Princip přiřazení

- Zápis programu pro přiřazení hodnot do proměnných *a* a *b* a následné přiřazení proměnné *b* do *a*.

Přiřazení hodnoty proměnné

```
1 int a = 10;
2 int b = 20;
3
4 a = b;
```

- Jaké jsou hodnoty proměnných *a* a *b*?

- | | |
|-------------------|-------------------|
| a. a = 20, b = 0 | f. a = 30, b = 0 |
| b. a = 20, b = 20 | g. a = 10, b = 30 |
| c. a = 0, b = 10 | h. a = 0, b = 30 |
| d. a = 10, b = 10 | i. a = 10, b = 20 |
| e. a = 30, b = 20 | j. a = 20, b = 10 |

Program vlastně „jen“ přesouvá a upravuje číselné hodnoty v paměti na základě definovaných podmínek!

Způsob výuky programování v BAB36PRGA

- Naší snahou je vybudovat zkušenost a rozvinout dovednost programování.
 - Programování vs. algoritmizace;
 - Programování je „řemeslo“, jak správně implementovat nějaký algoritmus.
 - **Jen funkční nestačí - program musí být i správně!** *Očekávaný vstup vs. co všechno může uživatel na vstup zadat.*
- Studijní zátěž je proto rozložena do výukové části semestru.
 - Úkoly na cvičení a termíny domácích úkolů.
- Systematické rozvíjení dovednosti programování v průběhu semestru považujeme za zásadní.

Typicky je na začátku semestru čas pro pochopní principů (čtení učebnice)!
- Bez znalosti konstruktů a základní příkazů nelze efektivně programovat.
- Vědět a umět použít (nikoliv „slepot“). *Nezávislost na našeptávači či Co-pilotu!*
 - Začínáme relativně jednoduchými úlohami k osvojení programovacích konstruktů a způsobu organizace zdrojového kódu. *Přehledný kód a schopnost se efektivně orientovat v kódu!*
 - *Úkoly jdou vždy realizovat s tím, co si řekneme na přednášce/cvičení.*

Řešení s pokročilejšími konstrukty může být elegantnější(kratší), nemusí však dodat potřebný vhled.
 - V prvních přednáškách pokrýváme nezbytné znalosti, které jsou dále prohlubovány.
 - Cvičení dopňují přednášky a dávají více prostoru pro praktické osvojení problematiky.
- Můžete volit praktický způsob vstřebávání znalosti programování z příkladů, který je vhodný doplnit **teoretickou přípravou z učebnic(e).**

Program je „recept“

- Program je „recept“ – posloupnost kroků (výpočtů) popisující průběh řešení problému.
- Programování je schopnost **samostatně**
 - **tvořit programy;**
 - **dekomponovat** úlohy na menší celky;
 - sestavovat z **dílčích částí větší programy** řešící komplexní úlohu.

BAB36PRGA – je příležitostí, jak se těmto schopnostem naučit!

Přehled přednášek

- Státní svátek*
- 01 - Úvod do programování v C *S. G. Kochan: kapitoly 1–3*
 - 02 - Základy programování (v C) *S. G. Kochan: kapitoly 2–5 a část 6*
 - 03 - Řídící struktury, výrazy a funkce *S. G. Kochan: kapitoly 4–6 a 12*
 - 04 - Pole, ukazatel, textový řetězec, vstup a výstup programu *S. G. Kochan: kapitoly 7, 10 a 11*
 - 05 - Ukazatele, paměťové třídy a volání funkcí *S. G. Kochan: kapitoly 8 a 11*
 - 06 - Struktury a uniony, přesnost výpočtů a vnitřní reprezentace číselných typů *S. G. Kochan: kapitoly 9, 14, 17 a Appendix B*
 - 07 - Standardní knihovny C. Rekurze. **(Základní vlastnosti jazyka C probrány.)**
 - 08 - Spojové struktury *S. G. Kochan: kapitola 16 a Appendix B*
 - 09 - Abstraktní datový typ (ADT) - zásobník, fronta, prioritní fronta
 - 10 - Paralelní programování, paralelní výpočty a synchronizační primitiva (semafory, zprávy a sdílená paměť)
 - 11 - Vícevláknové programování, modely aplikací, POSIX vlákna C11 vlákna
 - 12 - ANSI C, C99, C11 a rozdíly mezi C a C++. Úvod do C++ v příkladech
 - 13 - Rezerva - Rektorský den
 - 14 - Stručný úvod do C++ (v příkladech)

Přednáška nemusí být prezentace slidů – je očekávána interakce, řešení dotazů a diskuse problematických a náročnějších částí.

Podklady k přednášce jsou k dispozici před přednáškou podobně jako učebnice.

Domácí úkoly a další úlohy

- Samostatná práce s cílem osvojit si praktické zkušenosti.
- Jednotné zadání na přednášce a jednotný termín odevzdání.
- Odevzdání domácích úkolů prostřednictvím BRUTE.
<https://cw.felk.cvut.cz/brute>
 - Nahrání (upload) archivu s nezbytnými zdrojovými soubory.
 - Ověření správnosti implementace automatickými testy.
 - Penalizace za překročení počtu uploadů.

Odevzdávejte funkční kódy, nikoliv „pouze“ kódy, které projdou testy!

 - Detekce plagiátů
Cílem řešení úkolů je získat vlastní zkušenost!
- Úkoly jsou navrhovány tak, aby byly stihnutelné. Plánujte a hlídejte si čas, včas konzultujte.
- Klíčem k úspěšnému dokončení předmětu je samostatná práce a osvojení si technik a znalostí
Průběžná práce a řešení úkolů!
- Pokud něčemu nerozumíte, **ptejte se!**
Pokud chybujete, tak se učíte, pokud nechybujete, tak už to umíte!

Odevzdávání domácích úkolů – BRUTE

- **BRUTE** – Bundle for Reservation, Uploading, Testing and Evaluation
 - Formální kontrola – kompilace programu.
 - Testování funkčnost a správnosti – **kontrola výstupu pro daný vstup**.
 - Veřejné vstupy a odpovídající výstupy / neveřejné vstupy.
 - Před uploadem programu si program otestujete sami.
 - S využitím dostupných vstupů a výstupů.
 - Vytvoření vlastních vstupů a laděním programu.
 - Vytvoření vstupů **přiloženým generátorem vstupů**.
 - Ověření výstupu **přiloženým testovacím nebo referenčním programem**.
- Porozumění kódu a kontrola možných stavů.
 - **Pro každý řádek byste měli být schopni odpovědět proč tam je a co dělá!**
 - Pro **každou funkci nebo načtení vstupu** od uživatele analyzujte možné vstupní hodnoty nebo **návratové hodnoty funkcí!**
 - Pokud je z hlediska funkčnosti vstup nebo návratová hodnota zásadní, **proved'te kontrolu vstupu a/nebo příslušnou akci**, např. vypsání hlášení a ukončení programu.
Např. očekávaný vstup je číslo a uživatel zadá něco jiného.

Přehled domácích úkolů

- Domácí úkoly s povinným, **volitelným**, případně bonusovým zadáním. *52 h, bonus +28 h*
<https://cw.fel.cvut.cz/wiki/courses/b3b36prga/hw/start>
 0. HW 0 - První program *1 h*
 1. HW 1 - Načítání vstupu, výpočet a výstup
(**Kontrola přehlednosti kódu** – až -100% z dosažených bodů) *4 h*
Seznámení se s prostředím, psaním programů, jejich laděním, testováním a odevzdáváním. ~ 20–40 h
 2. HW 2 - Kreslení (ASCII art) (**Kontrola kódu** – až -100%) *5 h*
 3. HW 3 - Prvočíselný rozklad (**Kontrola kódu** – až -100%) *5 h, bonus +8 h*
 4. HW 4 - Caesarova šifra (**Kontrola kódu** – až -100%) *6 h, bonus +6 h*
 5. HW 5 - Hledání textu v souborech *6 h*
 6. HW 6 - Maticové počty (**Kontrola kódu** – až -100%) *6 h bonus +6 h*
 7. HW 7 - Kruhovná fronta v poli - *Dynamicky linkovaná knihovna* *6 h*
 8. HW 8 - Fronta spojovým seznamem s řazením *7 h*
 9. HW 9 - Vícevláknová aplikace s meziprocesovou komunikací. *6 h bonus +8 h*
- Podmínkou zápočtu je úspěšné odevzdání všech domácích úkolů.
- Odevzdání volitelného zadání je doporučeno (není částečné odevzdání).
Celkové body za povinné zadání **30b**, volitelné zadání **15b**, bonusové **20b**.

Úkoly a BRUTE

- Úkoly nejsou jen o odevzdání implementace, která projde BRUTE testy.
 - Cíl není odevzdat úkoly v BRUTE, je to prostředek ověření funkčnosti programu.
 - BRUTE je nástroj průběžné kontroly postupu a získávání znalostí.
 - Cíl je naučit se **samostatně programovat** funkční programy správně.
- Úkoly jsou především o **postupném získání zkušeností** s konkrétními konstrukty.
 - Všechna zadání úkolu už byla mnohokrát implementována, a zvládne to i generativní AI.
V předmětu máte příležitost porozumět programování v C prostřednictvím vlastní implementace úkolů. Odevzdání úkolu je prostředek dosažení cíle, nikoliv cíl samotný.
- Úkoly mají relativní obtížnost velmi podobnou.
 - Je důležité postupně samostatně řešit jednotlivé úkoly a osvojovat si dílčí dovednosti.
Absolutně jsou úlohy postupně náročnější a náročnější!
- Netrapte se s řešením příliš dlouho sami, ptejte se (discord), na cvičení nebo **konzultaci**.
- Úkoly HW1–HW4 a HW6 budou kontrolovány na správnost a přehlednost kódu.
 - Zaměřeno na konzistenci, čitelnost, a **modularitu** (rozdělní do funkcí).
Z hlediska tréninku a učení, i zdánlivě triviální program se snažte rozumně rozdělit na více funkcí.
 - **Motivace je netrávit příliš mnoho času implementací bez výrazného postupu.**

Hodnocení předmětu

Zdroj bodů	Maximum bodů	Přípustné minimum bodů
Domácí úkoly	45	} 35 †10 10
Bonusové úkoly	20	
Písemný zkuškový test	20	
Implementační zkouška	20	
Součet	105 bodů	

†V případě neúspěšné implementace a zkuškového testu na 13 a více bodů, je opakovaná zkouška pouze z implementační částí, a vice versa.

- **Zápocet:** alespoň 35 bodů a odevzdané všechny domácí úkoly **do 24.05.2024@23:59 CEST!**
- Předmět lze úspěšně ukončit **zápočtem a zkouškou.**
Zkuškové termíny (KOS) a <https://cw.fel.cvut.cz/wiki/courses/bab36prga/start>.
- Test a písemná zkouška – krátké stručné odpovědi prokazující porozumění problematice.
https://cw.fel.cvut.cz/wiki/courses/bab36prga/resources/exam_test
- **Implementační zkouška** – prokázání samostatně porozumět a napsat krátký program.
<https://cw.fel.cvut.cz/wiki/courses/bab36prga/resources/exam>

Komunikace související s PRGA

- Obracejte se na svého cvičícího dle cvičení, na které chodíte (jste přihlášení).
Případně na prg-teachers@fel.cvut.cz
- Komunikovat můžete elektronickou poštou (e-mail).
 - Pište ze své **fakultní adresy** (odesílatel).
 - **Do předmětu zprávy uvádějte zkratku předmětu PRG nebo PRGA.**
 - Kopii zprávy (Cc) posílejte též příslušnému vedoucímu cvičení (dle studijního programu).
 - V případě zásadních problému (např. týkajících se zápočtu) uvádějte do Cc též přednášečícího.
- Případně můžete využít **discord** kanálu. Time management - nečekejte okamžitou odpověď.
Pracujte soustředěně a užíjte si tvůrčí zápal.

Klasifikace předmětu

Klasifikace	Bodové rozmezí	Hodnocení	Slovní hodnocení
A	≥ 90	1	výborně
B	80–89	1,5	velmi dobře
C	70–79	2	dobře
D	60–69	2,5	uspokojivě
E	50–59	3	dostatečně
F	<50	4	nedostatečně

- Včasné odevzdáním všech domácích úkolů s povinným a **volitelným** zadáním (**45 bodů**).
- Bonusové úlohy (**20 bodů**).
- Písemná zkouška (**20 bodů**) . 15 a více bodů je velmi slušný výsledek!
- Implementační zkouška (**20 bodů**).
- **95 bodů** a více (A – výborně), **76 bodů** (C – dobře) – (20% ztráta).
- Body jsou indikátorem průběžných výsledků.

Vývojové prostředí

- Počítačové laboratoře - Ubuntu se síťovým bootováním a domovskými adresáři (NFS v4).
Přenos a synchronizace souborů – ownCloud, SSH, gdrive, sharepoint
- Doporučený operační systém - Ubuntu-based, Pop OS!, Win s WSL(2) ideální s nativní VS Code.
Přímocará instalace potřebných programů.
- Překladače **gcc** a **clang**, sestavení **make** (GNU make). <https://gcc.gnu.org> a <http://clang.llvm.org>
- **Visual Studio Code** (VSC) - <https://code.visualstudio.com/>
- Editor – **gedit**, **atom**, **sublime**, **vim** – <https://atom.io/>, <http://www.sublimetext.com/>
<http://www.root.cz/clanky/textovy-editor-vim-jako-ide>
Pokud programovat umíte, investuje čas do efektivního ovládání editoru, např. **vim**.
- C/C++ vývojová prostředí – **WARNING: Do Not Rely on an IDE.**
<http://c.learnthecodethehardway.org/book/ex0.html>
 - **CLion**, NetBeans 8.0 (C/C++), Eclipse-CDT – <https://www.jetbrains.com/clion>
 - **Geany**, Code::Blocks, CodeLite <https://www.geany.org/>, <http://www.codeblocks.org>, <http://codelite.org>
 - **Nejdříve porozumějte principům**, nakonfigurujte nástroje a programování zefektivněte.
- Odevzdávání domácích úkolů BRUTE <https://cw.felk.cvut.cz/brute>.
BRUTE – Bundle for Reservation, Uploading, Testing and Evaluation.

Služby akademické sítě – FEL, ČVUT

- <http://www.fel.cvut.cz/cz/user-info/index.html>
- Diskové úložiště ownCloud – <https://owncloud.cesnet.cz>
- Zasílání velkých souborů – <https://filesender.cesnet.cz>
- Rozvrh a termíny – FEL Portal – <https://portal.fel.cvut.cz>
- FEL Google Account – autentizovaný přístup do [Google Apps for Education](#)
Více viz <http://google-apps.fel.cvut.cz/>.
- Gitlab FEL – <https://gitlab.fel.cvut.cz/>
- Přístup k informačním zdrojům (IEEE Xplore, ACM, Science Direct, Springer Link).
- Akademické a kampusové licence. <https://download.cvut.cz>
- Národní Gridová Infrastruktura MetaCentrum. <http://www.metacentrum.cz/cs/index.html>
- RCI Cluster. <https://login.rci.cvut.cz>