

Informace o předmětu, organizace

B0B99PRPA – Procedurální programování

A8B14ADP – Algoritmizace a programování

Stanislav Vítek

Katedra radioelektroniky
Fakulta elektrotechnická
České vysoké učení v Praze

Předmět a lidé

- Webové stránky předmětu

<https://cw.fel.cvut.cz/wiki/courses/b0b99prpa/start> ↗

- Přednášející a garant předmětu

- Stanislav Vítek, viteks@fel.cvut.cz

<http://mmtg.fel.cvut.cz/personal/vitek/> ↗

- Cvičící

- Martin Mudroch, mudromar@fel.cvut.cz
 - Ondřej Nentvich, nentvond@fel.cvut.cz
 - Václav Navrátil, vaclav.navratil@fel.cvut.cz
 - Josef Krška, josef.krska@fel.cvut.cz
 - Lukáš Liebzeit, liebzluk@fel.cvut.cz
 - Filip Žourek, zourefil@fel.cvut.cz

- Konzultace

- MS Teams, individuálně po domluvě

Cíle předmětu

- **Motivovat k programování**

- Programování je klíčová dovednost, která může hrát rozhodující roli na trhu práce

- **Naučit se algoritmizovat**

- Formulace problému a návrh řešení
- Rozklad problému na dílčí úlohy
- Identifikace opakujících se vzorů

- **Získat zkušenosti s programováním**

- Základní programovací konstrukce

Proměnné, cykly, podmínky, datové struktury a jednodušší algoritmy

- Programovací jazyk C, řada principů obecně použitelných

Cvičení, domácí úkoly, hledání chyb, práce s dokumentací, test

Programátorovi nestačí perfektní znalost programovacího jazyka, ale především musí vědět, jak vůbec danou úlohu řešit.

Organizace a hodnocení předmětu

- **Studijní výsledky**

- Průběžná práce v semestru – domácí úkoly a testy
- Implementační test

- **Docházka**

- Přednášky jsou nepovinné, ale snad přínosné a zábavné
- Cvičení jsou povinná, možné dvě omluvené absence
- Na cvičení se očekává aktivní účast při řešení příkladů

Na cvičení je třeba se **přípravit**, nejlépe návštěvou přednášky a studiem podkladů (řešené příklady)

- **Řešení problémů**

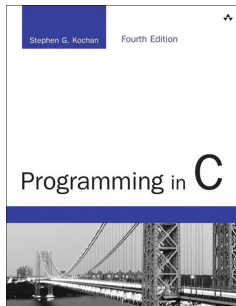
- Obracejte se na svého cvičícího
- Při komunikaci e-mailem pište vždy ze své fakultní adresy
- Do předmětu zprávy uvádějte zkratku předmětu PRPA
- V případě zásadních problémů uvádějte do CC též přednášejícího

Zdroje a literatura



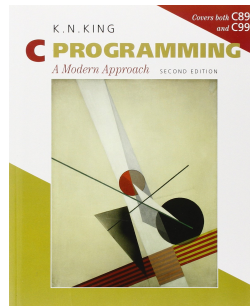
Pavel Herout
Učebnice jazyka C

Kopp, 2011
ISBN 978-80-7232-383-8



Stephen G. Kochan
Programming in C

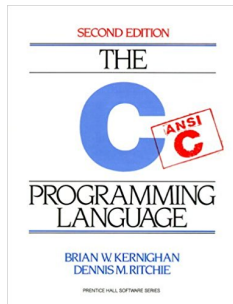
Addison-Wesley 2014
ISBN 978-0321776419



K. N. King
C Programming: A Modern Approach

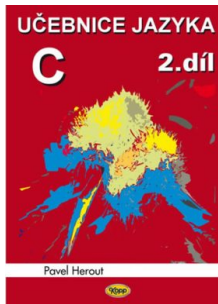
W. W. Norton & Company 2008
ISBN 860-1406428577

Zdroje a literatura



Brian W. Kernighan
Dennis M. Ritchie
[The C Programming Language \(ANSI C\)](#)

Prentice Hall, 1988
ISBN 978-0131103627



Pavel Herout
[Učebnice jazyka C – 2. díl](#)

Kopp, 2008
ISBN 978-80-7232-367-8



Peter van der Linden
[Expert C Programming: Deep C Secrets](#)

Prentice Hall, 1994
ISBN 978-0131774292

Domácí úkoly

- Samostatná práce, praktické zkušenosti s programováním
- Náročnost domácích úkolů se postupně zvyšuje
- Odevzdání domácích úkolů v systému **BRUTE** [↗](#)
- Cílem řešení úkolů je získat **vlastní** zkušenost
 - Úkoly mají formativní charakter
 - Neopisujte, nespolehejte na AI – škodíte především sobě
 - Automatická kontrola plagiátů u všech odevzdaných řešení
 - každý s každým
 - každý s řešením z minulých let (pokud je podobný příklad)
 - u podezřelých případů provedeme manuální kontrolu
 - V případě odhalení jsou potrestáni **oba** účastníci incidentu



Pokud něčemu nerozumíte, ptejte se!

Přehled domácích úkolů

HW01	– První program, Hello PRPA! (povinný)	(1)
HW02	– Načítání vstupu, výpočet a výstup (povinný)	(4)
HW03	– První cyklus (volitelný)	(4)
HW04	– RLE kodér (povinný)	(5)
HW05	– Kreslení (volitelný)	(6)
HW06	– Caesarova šifra (povinný)	(5)
HW07	– Maticové počty (volitelný)	(9)
HW08	– Zpracování číselné řady (povinný)	(5)
HW09	– Analýza textového souboru (povinný)	(5)
HW10	– Kruhová fronta v poli (volitelný)	(6)

Celkem lze získat až **25b** za povinné HW a dalších **25b** za nepovinné.

Kontrola domácích úkolů – odevzdávací systém BRUTE

- Formální kontrola – kompilace programu
- Testování funkčnosti 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
 - Vytvořením vlastních vstupů a laděním programu
- Penalizace za překročení počtu uploadů
- Detekce plagiátů
- Porozumění kódu a kontrola možných stavů
 - 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, proveďte kontrolu vstupu a/nebo příslušnou akci, např. vypsání hlášení a ukončení programu

Např. očekávaný vstup je číslo a uživatel zadá něco jiného.

Implementační test

- Prokázání základních kompetencí v programování
- Řešení jednoho příkladu v čase maximálně 240 minut
- Implementace probíhá na školních počítačích
- Bez přístupu na internet
- Povoleny papírové a elektronické materiály
- Automatické vyhodnocení pomocí BRUTE

Hodnocení

Zdroj bodů	Maximum	Minimum
Domácí úkoly povinné	25	doporučené: 35
Domácí úkoly nepovinné	25	
Testy v semestru	30	15
Implementační test	30	10
Součet	110	

- Za práci v semestru **je doporučeno získat nejméně 35 bodů**.
- Všechny povinné domácí úkoly musí být odevzdány ve **funkční podobě**.
- Nejpozdější odevzdání **11.1.2026**.
- Počet bodů je ekvivalentní procentuálnímu hodnocení klasifikační stupnice ČVUT.

A (≥ 90), **B** (80 – 89), **C** (70 – 79), **D** (60 – 69), **E** (50 – 59), **F** (< 50)