

Taschenrechner

Allgemeine Anmerkungen zur Aufgabenstellung:

- Beantworten Sie die Kontrollfragen in TUWEL bevor Sie mit der Übung beginnen.
- Stellen Sie während der Übung laufend sicher, dass Ihr Programm kompilierbar ist und richtig funktioniert!
- Die Abgabe erfolgt durch Uploaden Ihres vollständigen `Code::Blocks`-Projektordners als zip-Datei (`Matrikel-Nr_UE2.zip`) in TUWEL. Anschließend melden Sie sich für das Abgabegespräch in TUWEL an. Dieses erfolgt über die Telekonferenz-App Zoom (www.zoom.us). Stellen Sie sicher, dass diese App rechtzeitig vor Beginn des Abgabegesprächs auf Ihrem PC installiert ist und dass Sie Ihr aktuelles Übungsprogramm in `Code::Blocks` für die Abgabe geöffnet haben. Testen Sie bitte auch die korrekte Funktionalität Ihrer Videokamera, da Sie zu Beginn Ihres Abgabegesprächs Ihren Studentenausweis herzeigen müssen und auch während des Gesprächs eine aktive Videoverbindung erforderlich ist.
- Beachten Sie, dass Upload und Terminauswahl für das Abgabegespräch erst möglich sind, wenn Sie 80% der Kontrollfragen richtig beantwortet haben. Diese müssen, genauso wie die Übung, bis spätestens **26.04.2020 23:59** durchgeführt werden.
- Inhaltliche Fragen stellen Sie bitte im zugehörigen Forum in TUWEL. Das Betreuungsteam steht Ihnen in den jeweiligen Übungswochen von Mo.-Fr. in der Zeit von 09:00-17:00 zur Verfügung. Außerhalb dieser Zeiten sind wir auch bemüht Ihre Anfragen möglichst zeitnah zu beantworten.
- Für eine positive Beurteilung Ihrer Abgabe **muss** diese kompilierbar sein, Ihr Programm fehlerfrei terminieren und die in der Angabe angeführten Funktionen mit den vorgegebenen Funktionsköpfen enthalten. Des Weiteren müssen die Funktionen hinreichend getestet sein. Sie müssen in der Lage sein, Ihr Programm beim Abgabegespräch zu erklären!

Folgende Hilfsmittel stehen Ihnen in TUWEL zur Verfügung:

- Alle bisherigen Referenzbeispiele sowie Vorlesungsfolien
- Interaktive Jupyter-Notebooks auf prog1.iue.tuwien.ac.at
- Das Buch zur LVA "Programmieren in C" (Robert Klima, Siegfried Selberherr)
- Kurzanleitung für die Entwicklungsumgebung `Code::Blocks`

Hinweis: Schreiben Sie Ihr Programm selbstständig! Das Kopieren der Lösung stellt einen groben Verstoß gegen die Durchführungsbestimmungen der LVA (siehe LVA-Beschreibung im TISS) dar, welcher zu einer negativen Beurteilung der Übung führt.

Einleitung

In dieser Übung werden Sie einen einfachen Taschenrechner in der Programmiersprache C implementieren. Sie lernen dabei die Mathematik-Bibliothek zu benutzen und eine Ablaufsteuerung mittels Verzweigungen zu realisieren.

Sie benötigen für diese Übung folgende neue Punkte aus dem Stoff-Gebiet:

- Numerische Datentypen `char`, `long`, `double` (Kapitel 6)
- Mathematische Funktionen `exp()`, `sin()`, `cos()`, ... (Kapitel 6.2.4)
- Eingabe/Ausgabe mit Platzhaltern: `printf()`, `scanf()` (Kapitel 7)
- Ausdrücke (Kapitel 8)
- Selektionen: `if`, `else`, `switch`, `case` (Kapitel 9)



Teil A

- ✕ Arithmetische Ausdrücke sollen in der üblichen Reihenfolge eingegeben werden. Lesen Sie also der Reihe nach den ersten Operanden, den Operator und den zweiten Operanden ein. Anschließend geben Sie das berechnete Ergebnis aus. Ein typischer Programmablauf könnte zum Beispiel so aussehen:

```
Operand:  7.2
Operator:  +
Operand:  5.3
=====
Ergebnis: 12.500000
```

Der Taschenrechner soll die vier Grundrechnungsarten Addition (+), Subtraktion (-), Multiplikation (*) und Division (/) unterstützen. Überprüfen Sie, ob die eingegebenen Operanden gültig sind und vermeiden Sie eine Division durch Null.



Teil B

- ✕ Erweitern Sie nun Ihren Taschenrechner um verschiedene wissenschaftliche Funktionen unter Zuhilfenahme der Mathematik-Bibliothek. Folgende Tabelle zeigt die Funktionen, die Ihr Taschenrechner beherrschen soll:

Potenzfunktion	(p)	$z = \text{pow}(x, y)$
Cosinus	(c)	$y = \cos(x)$
Logarithmus	(l)	$y = \log(x)$

Die Eingaben

```
Operand:  2
Operator:  p
Operand:  3
```

und

```
Operand:  30
Operator:  c
```

bedeuten 2^3 beziehungsweise $\cos(30^\circ)$. Funktionen werden also durch Eingabe eines Zeichens ausgewählt. Winkel werden in Grad eingegeben und müssen in Radiant an die Winkelfunktionen übergeben werden ($180^\circ = \pi$ rad). Fangen Sie Operanden, die nicht im Definitionsbereich der jeweiligen Funktion liegen, durch eine Fehlermeldung ab.



Teil C

- ✕ Erweitern Sie Ihren Taschenrechner um eine Funktion, die eine eingebene Zahl auf eine bestimmte Anzahl von Stellen rundet. Lassen Sie den Benutzer die Anzahl der Nachkommastellen N , auf welche gerundet werden soll, eingeben. Anschließend soll das Ergebnis mit $N+3$ Nachkommastellen ausgegeben werden. Verwenden Sie für die Implementierung des Rundens die Funktion `floor` aus der Mathematikbibliothek. Die Ausgabe der Rundungsoperation kann beispielsweise wie folgt aussehen:

```

Operand:          2.4565656665
Operator:         r
Nachkommastellen: 0
=====
Ergebnis:        2.000

```

oder

```

Operand:          2.4565656665
Operator:         r
Nachkommastellen: 3
=====
Ergebnis:        2.457000

```

- ✘ Erweitern Sie das Programm so, dass mit dem letzten Ergebnis weitergerechnet werden kann. Bei der Operation Runden soll nur das gerundete Ergebnis ausgegeben, jedoch mit dem nicht gerundeten Ergebnis weitergerechnet werden. Zum Beenden der Rechnung geben Sie als Operator das Zeichen „.“ ein.

Beispielablauf:

```

Operand:  7.2
Operator: +
Operand:  5.3
=====
Ergebnis: 12.500000

```

```

Operator: *
Operand:  2.0
=====
Ergebnis: 25.000000

```

```

Operator: .
=====
Programmende

```