



## Programmierpraktikum Technische Informatik (C++)

### Aufgabe 1

#### Hinweise

Aufgaben müssen in Einzelarbeit bearbeitet werden.

Die abgegebene Lösung muss ohne Warnungen kompilieren, ansonsten droht Punktabzug.

**Abgabe: Stand des Git-Repositories am 18.4.2023 um 9 Uhr.**

#### Teilaufgabe 1 (0 Punkte)

Richten Sie sich eine Programmierumgebung ein! Lesen Sie dazu die Vorlesungsfolien durch und installieren Sie die zur Verfügung gestellte VirtualBox-Appliance. Diese Aufgabe wird nicht kontrolliert, ist aber für die Bearbeitung der folgenden Aufgaben notwendig.

#### Teilaufgabe 2 (2 Punkte)

Richten Sie Ihr git-Repository ein! Dazu sind die folgenden Arbeitsschritte notwendig:

Hinweise: Schauen Sie sich hierzu nochmal das git-Kapitel (besonders "Einrichten der Programmierumgebung I/II" und "Verwendeter Git Workflow") in den Vorlesungsfolien an!

- a) Konfigurieren Sie git mit ihrem Namen und Email-Adresse!
- b) Suchen Sie in Ihren E-Mails nach der Mitteilung über das Ihnen zugewiesene Repo. Falls Sie keine finden, bitte fragen Sie nach.
- c) Klonen Sie das Ihnen in der ersten Vorlesung zugewiesene Repository `git@gitlab.uni-hannover.de/ppti/2023/<repo>.git` und checken Sie `main` aus!
- d) Unter `git@gitlab.uni-hannover.de/ppti/2023/ppti-common.git` finden Sie das Aufgabenrepository. In diesem werden für die einzelnen Aufgaben jeweils Codevorgaben veröffentlicht. Machen Sie das Aufgabenrepository mit `git remote add common ...` dem lokalen Repository bekannt!
- e) Fetchen Sie das Aufgabenrepository und mergen Sie dessen `main`-Branch in ihren eigenen. **Geht dies mit einem einzigen git-Befehl?**
- f) Pushen Sie ihren `main`-Branch nach `origin`!



- g) Überprüfen Sie mithilfe des Werkzeuges gitk, ob Ihr push erfolgreich war und die Änderungen auf dem Server liegen! **Notieren Sie die in gitk angezeigte SHA1-ID des entsprechenden Commits!**

Bitte erzeugen Sie eine Textdatei, in der Sie die Fragen in den Teilen e) und g) schreiben!

### Teilaufgabe 3 (2 Punkte)

In Ihrem main-Branch sollte unter Aufgabe01 eine Datei main.cpp liegen. Versuchen Sie diese zu kompilieren! Das Kommando dazu lautet:

```
g++ -std=c++20 -Wall -o main main.cpp
```

Sie werden feststellen, dass der Code einige Fehler aufweist. Beheben Sie diese! Das kompilierte Programm können Sie mit ./main starten. Checken Sie das korrigierte Programm in Ihrem Repository auf dem git-Server ein (commit und push)! Vergessen Sie nicht, dass Sie Änderungen mit git add oder git commit -a dem Index hinzufügen müssen.

In korrekter Form wartet das Programm auf die Eingabe eines Namens und gibt, nachdem diese erfolgt ist, zehn Zeilen mit dem Text hello gefolgt vom eingegebenen Namen auf der Konsole aus. Abweichungen von diesem Verhalten sind Fehler im Code!

### Teilaufgabe 4 (1 Punkt)

Schreiben Sie einen einfachen Taschenrechner, der die Rechenoperationen Addition und Subtraktion anbietet! Der Benutzer wird aufgefordert, zwei Zahlen und eine Rechenoperation add oder sub einzugeben. Das Programm soll die Eingaben verarbeiten und anschließend das Ergebnis auf der Konsole ausgeben.

Passen Sie die Datei /calc.cpp unter Aufgabe01/Calculator entsprechend an und testen Sie Ihr Ergebnis mit dem Kommando g++ -std=c++20 -Wall -o calc calc.cpp!

Das Programm können Sie mit ./calc starten. Checken Sie das Programm in Ihrem Repository auf dem git-Server ein (commit und push)!

**Hinweis:** In C++ können Strings, anders als in Java, direkt mit den Vergleichsoperatoren == auf Gleichheit bzw. != auf Ungleichheit geprüft werden.

Die Bedienung des Programms könnte wie folgt aussehen:

Number1: 2

Number2: 3

Operation: add

Result: 5