In this exercise sheet we work with precondition-postcondition pairs and the Hoare proof system for our Boostan language.

Submit your solution by uploading it as PDF in ILIAS.

**Exercise 1: Precondition - Postcondition**
4 Points

Consider the following precondition-postcondition pairs. Which of them are satisfied by all program statements \( st \) and all formulas \( \varphi \)?

(a) \( \{ \text{true} \} \) \( st \) \( \{ \varphi \} \)

(b) \( \{ \text{false} \} \) \( st \) \( \{ \varphi \} \)

(c) \( \{ \varphi \} \) \( st \) \( \{ \text{true} \} \)

(d) \( \{ \varphi \} \) \( st \) \( \{ \text{false} \} \)

If a precondition-postcondition is satisfied by all program statements \( st \) and all formulas \( \varphi \), then explain why. If a precondition-postcondition is not satisfied by some program statement \( st \) and some formulas \( \varphi \), then give a counterexample.

**Exercise 2: Assignment Axiom**
1 Point

Find some program \( P \) whose code is a single assignment statement of the form \( x := \text{expr} \); and some formula \( \varphi \) such that \( P \) does not satisfy the precondition-postcondition pair \( (\{ \varphi \}, \{ \varphi \land x = \text{expr} \}) \).

The motivation of this exercise is the following. In the lecture we have seen the assignment axiom of the Hoare proof system.

\[
\begin{align*}
(\text{assign}) & \quad \{ \varphi[x \mapsto \text{expr}] \} \; x := \text{expr}; \; \{ \varphi \}
\end{align*}
\]

This rule is not very intuitive because the precondition is obtained as a modification of the postcondition. One may wonder if the following proof rule could be an alternative.

\[
\begin{align*}
(BadAss) & \quad \{ \varphi \} \; x := \text{expr}; \; \{ \varphi \land x = \text{expr} \}
\end{align*}
\]

The result of this exercise should hint that the (BadAss) proof rule cannot be used as an axiom in a proof system whose goal is the derivation of valid Hoare triples.
Exercise 3: Hoare Proof System

Is there a program that can swap the values of two variables without using a temporary variable? In this exercise we will consider such a program and prove that the program indeed has this property.

Consider the Boostan program $P_{\text{swap}} = (V, \mu, \mathcal{T})$ with $V = \{a, b, x, y\}$, $\mu(a) = \mu(b) = \mu(x) = \mu(y) = \mathbb{Z}$, and $\mathcal{T}$ a derivation tree for the program code shown below.

$$
\begin{align*}
x & := x + y; \\
y & := x - y; \\
x & := x - y;
\end{align*}
$$

Use the Hoare proof system to show that $P$ satisfies the precondition-postcondition pair $(\{x = a \land y = b\}, \{x = b \land y = a\})$.

Exercise 4: Programming in Boogie

Using the Boogie language, implement a procedure with signature

$$\text{procedure square}(x : \text{int}) \text{ returns } (z : \text{int})$$

that takes a (mathematical) integer $x$ and, if it is greater or equal 0, computes and returns the square $z = x^2$. The algorithm may only make use of addition and subtraction, but not use multiplication, division or modulo.

You can use the Boogie interpreter Boogaloo to test your program. A user manual is available online.

Exercise 5: Hoare Logic Proof

Consider the following Boostan program $P = (V, \mu, \mathcal{T})$ with $V = \{i, j, x, y\}$, $\mu(i) = \mu(j) = \mu(x) = \mu(y) = \mathbb{Z}$, and $\mathcal{T}$ a derivation tree for the program code shown below.

$$
\begin{align*}
x & := i; \\
y & := j; \\
\text{while} (x \neq 0) \{ \\
& \quad x := x - 1; \\
& \quad y := y - 1;
\}
\end{align*}
$$

Give a Hoare logic proof showing that $\{\text{true}\} \mathcal{T} \{i = j \rightarrow y = 0\}$ is a valid Hoare triple.

---