

Quiz 9

The Art of Computer Programming

5 January 2026

Full name:

You have 10 minutes to answer this quiz, directly on this sheet of paper. No electronic devices or material of any kind is allowed. Do not forget to add your name above. Every question amounts to 2 points; the quiz is graded out of 10 points.

Questions

Q1. Circle the correct statement about **building correct programs**:

- a. Testing can replace specifications and reasoning.
- b. Debugging is mostly random trial-and-error until the program works.
- c. Specifications, reasoning, testing, and debugging play different roles and complement each other.
- d. A program is correct as soon as it runs without crashing once.

Q2. Preconditions and postconditions. Consider a method `withdraw(amount)` that removes money from an account which has some (non-negative) `balance`. Which specification is the best match?

- a. Pre: `amount ≥ 0`. Post: `balance_new = balance_old - amount`.
- b. Pre: `amount > 0` and `balance ≥ amount`. Post: `balance_new = balance_old - amount`.
- c. Pre: `balance > 0`. Post: `balance_new > balance_old`.
- d. Pre: none. Post: `balance_new = balance_old`.

Q3. Loop invariant (concrete). We want to compute the maximum of a Python list `T`:

```
m = T[0]
for i in range(1, len(T)):
    if T[i] > m:
        m = T[i]
```

Circle the best loop invariant at the start of each iteration (with current index `i`):

- a. `m` is equal to `T[i]`.
- b. `m` is the maximum of the elements of `T` among those at position $0 \dots i - 1$.

- c. The Python list `T` is sorted.
- d. `i` never changes during the loop.

Q4. Assertions. Consider the following Python code:

```
def withdraw(balance, amount):  
    assert amount > 0  
    return balance - amount  
  
print(withdraw(10, 0))
```

What happens when this program is run?

- a. It prints 10.
- b. It prints 0.
- c. It raises an assertion failure before printing a number.
- d. It prints -10.

Q5. Testing vs debugging. Write a *single* unit test in **pytest style** (i.e., as a single function that should **assert** that everything goes fine) that checks that a transfer preserves total balance in the following API:

```
# methods:  
#   add_account(name, balance)  
#   transfer(src, dst, amount)  
#   getBalance(name)
```

Your test should:

- create two accounts "A" and "B";
- perform one transfer;
- assert that the sum of balances is unchanged.