



# CBDE<sup>Q&As</sup>

BTA Certified Blockchain Developer - Ethereum

## Pass Blockchain CBDE Exam with 100% Guarantee

Free Download Real Questions & Answers **PDF** and **VCE** file from:

<https://www.pass4itsure.com/cbde.html>

100% Passing Guarantee  
100% Money Back Assurance

Following Questions and Answers are all new published by Blockchain  
Official Exam Center

-  **Instant Download** After Purchase
-  **100% Money Back** Guarantee
-  **365 Days** Free Update
-  **800,000+** Satisfied Customers



**QUESTION 1**

The difference between `address.send()` and `address.transfer()` is:

- A. `.send` returns a Boolean and `.transfer` throws an exception on error. Both just forward the gas stipend of 2300 gas and are considered safe against re-entrancy.
- B. `.send` throws an exception and `.transfer` returns a Boolean on error. Both just forward the gas stipend of 2300 gas and considered safe against re-entrancy
- C. `.send` returns a Boolean and `.transfer` throws an exception on error. `.send` is considered dangerous, because it sends all gas along, while `.transfer` only sends the gas stipend of 2300 gas along
- D. `.send` and `.transfer` are both considered low-level functions which are dangerous, because they send all gas along. It's better to use `address.call.value()` to control the gas-amount.

Correct Answer: A

---

**QUESTION 2**

Multi-Line Comments in Solidity are:

- A. working with either `//` or `///`
- B. working with `/*` comment `*/` or `/** @.. natspec style */`
- C. not possible, all comments must be single-line.

Correct Answer: B

---

**QUESTION 3**

In order to implement an ERC20 token contract, you'd need at least to implement the following functions and events in order to fulfill the interface requirements:

- A. `totalSupply()`, `balanceOf(address)`, `allowance(address,address)`, `transfer(address,uint256)`, `approve(address,uint256)`, `transferFrom(address,address,uint256)`. Events: `Transfer(address,address,uint256)`, `Approval(address,address,uint256)`
- B. `name()`, `symbol()`, `totalSupply()`, `balanceOf(address)`, `ownerOf(uint26)`, `approve(address,uint256)`, `takeOwnership(uint256)`, `transfer(address,uint256)`, Events: `Transfer(address,address,uint256)`, `Approval(address,address,uint256)`

Correct Answer: A

---

**QUESTION 4**

Smart Contracts can be written in:



- A. Java, C++, Solidity and JavaScript, because the Ethereum Blockchain is completely language agnostic and cross compilers exist for every major language.
- B. Solidity, Viper, LLL and Serpent, because those are high level languages that are compiled down to bytecode.
- C. Solidity and JavaScript, because those are the official first implementations for Distributed applications and the Blockchain supports those languages fully.

Correct Answer: B

---

#### QUESTION 5

If you need more fine-grained functionality than solidity offers out of the box:

- A. you can incorporate inline-assembly to get better controls.
- B. you have to import pre-compiled assembly files which are then hard-copied into the bytecode of the compiled solidity file.
- C. you can use Viper, the experimental assembly like language specifically to offer more flexibility.

Correct Answer: A

[CBDE PDF Dumps](#)

[CBDE Practice Test](#)

[CBDE Brindumps](#)