End-Term Examination
CS578: Blockchain Technology: A S/W Perspective
Full Marks: 70 points Time: 3 hours

(Make assumptions whenever necessary)

1. **Choose correct answers from the given list (There may exist multiple correct answers in the list. There is no partial marking if you choose proper subset of the correct answers.)**

[10*2=20 points]

a. Which of the following statements are true?
   i. Go language supports only Imperative Programming Paradigm
   ii. Go language supports only Object-Oriented Programming Paradigm
   iii. Go language is a Multi-paradigm Programming Language
   iv. All of the above

b. What powers the Ethereum Virtual Machine?
   i. Gas
   ii. Ether
   iii. Bitcoin
   iv. Block Rewards

c. Solidity language is a
   i. Statically typed programming language
   ii. Dynamically typed programming language
   iii. Object-oriented high level language
   iv. All of the above

d. How would machine learning help in smart contract vulnerability analysis?
   i. Always improves precision
   ii. Helps to guarantee Soundness
   iii. May reduce computational cost
   iv. All of the above

e. In Paxos, a node can have only one role at a time among the three roles.
   i. True
   ii. False

f. At least how many total nodes is required in Tandermint Byzantine Fault Tolerance System to reach a consensus if 24 faulty nodes exist in the system?
   i. 72
   ii. 73
   iii. 48
   iv. 49

g. State machine replication-based consensus is used over permissioned blockchains. Select the suitable reason(s)?
   i. The network is closed, and the nodes know each other, hence state replication is possible among the known nodes
   ii. No need to spend power, time, or bitcoin
   iii. Machines can behave maliciously, hence consensus in required
   iv. State machine replication-based consensus is not recommended to use over permissioned blockchains

h. An orphan block is only created when 51% attack is successful.
   i. True

ii. False

i. Parity Multisig Wallet Hack occurs due to following reason
   i. Use of delegatecall
   ii. Use of tx.origin
   iii. Use of block-timestamp
   iv. None of the above

j. Which of the following statements are false?
   i. EVM is a stack-based machine
   ii. Smart contract's address is computed using the hash of the public key of sender who has deployed it
   iii. There may exist more than one fallback function in a Solidity smart contract
   iv. In a solidity smart contract, there may exist more than one constructor with different parameter list.

2. Explain re-enterancy attack in smart contract, using a suitable Solidity code example. Why transaction ordering dependences and block-timestamp dependences are considered vulnerable in Solidity Smart Contract (explain with suitable example in each of the two cases)?

[10+10=20 points]

3. Write the syntax of IMP language. Define formally the operational and the denotational semantics of the conditional statement. What is the basic difference between the semantic descriptions under these two semantic models?

[3+4+3=10 points]

4. Given a set of concrete states $\Sigma = \{\sigma_1, \sigma_2\}$ where $\sigma_1 = \langle x \to 5, y \to -7 \rangle$ and $\sigma_2 = \langle x \to 4, y \to 4 \rangle$. Compute the **denotational semantics** of the following statement w.r.t. Σ: *if(x==y) then x=8 else x=-5 endif.*

[8 points]

5. Consider the Raft replication system described in "In Search of an Understandable Consensus Algorithm" proposed by Ongaro and Ousterhout. Suppose you have a five-server Raft system. Here's the state of the servers' logs. The notation T.N means the Nth log entry from term T. The servers are about to choose a new leader.

[3*4= 12 points]

S1: 3.1 4.1
S2: 3.1 3.2
S3: 3.1 5.1
S4: 3.1 5.1 5.2
S5: 3.1

- Could any replica have already executed the operation in log entry 5.1? If yes, explain how this could have happened. If no, explain why it could not have happened.
- Could operation 3.2 be committed in the future? (Assume that the client does not re-transmit the operation.) If yes, how could that happen? If no, why not?
- Could operation 4.1 be committed in the future? (Assume that the client does not re-transmit the operation.) If yes, how could that happen? If no, why not?