上海市高等学校信息技术水平考试（二三级）

《区块链技术及应用》考试大纲

（2022 年版）

一、考试性质

上海市高等学校信息技术水平考试是上海市全市高校统一的教学考试，是检测和评价高校信息技术基础教学水平和教学质量的重要依据之一。该项考试旨在规范和加强上海高校的信息技术基础教学工作，提高学生的信息技术应用能力。考试对象主要是上海市高等学校在校学生。考试每年举行一次，通常安排在当年的十月下旬、十一月上旬的星期六或星期日。凡考试成绩达到合格者或优秀者，由上海市教育委员会颁发相应的证书。

本考试由上海市教育委员会统一领导，聘请有关专家组成考试委员会，委托上海市教育考试院组织实施。

二、考试目标

区块链作为一种数据存储模型，融合了一系列计算机相关技术，包括区块链底层技术如网络通信、密码编码学、多方安全计算、零知识证明、分布式存储、P2P 网络、隐私保护、智能合约、信息安全等；同时区块链的工程领域实现也提出了性能、分片、跨链、公有/联盟/授权链、链基础设施(BaaS, Blockchain as a Service)等系列相关技术；同时，区块链与人工智能、大数据、5G、物联网等前沿信息技术的深度融合，将极大地推动多学科间的集成创新和融合应用。

《区块链技术及应用》的考试目标是考核学生对分布式账本（区块链）基础知识的理解，针对特定场景进行分析并开发分布式应用的综合能力；考核学生对分布式账本（区块链）模式的理解，结合领域需求应用区块链技术优化业务流程的综合能力。考试内容涵盖从相关理论知识到基本方法的应用实践，要求学生熟练掌握区块链基础知识，理解关键技术，具备分析解决“区块链+”实际场景问题的能力，可以提出解决方案和决策建议，能够设计区块链项目的架构并进行部署、调整，具备进行动手配置、开发的能力。

三、考试内容和要求

知识领域	知识单元	知识点	要求
分布式账本（区块链）基础知识	基本概述	概念与定义	理解
		历史与现状	理解
		结构与模型（公有、私有、联盟）	理解
		系统分层	理解
	系统能力	多方共同维护	理解
		去中心化	理解
		不可篡改	理解
		匿名性	理解
		账本公开性	理解
	典型区块链项目	比特币	理解
		以太坊	理解
		超级账本	理解
		EOS	理解
		国产联盟链与 BaaS	知道
	密码学基础与编码	非对称加密（公私钥体系）	知道
		椭圆曲线	理解
		哈希函数	理解
		默克尔树	理解
		数字签名	理解
		特殊编码	理解
	激励机制和策略	经济博弈	理解
		代币模型	理解
		交易费	理解
	共识算法	共识系统基础	理解
		主流共识算法 PBFT/Pos/PoW 等	掌握
	安全与隐私保护	安全目标：数据安全、共识安全、智能合约安全、内容安全、隐私保护	理解
		区块链安全性问题（数据层安全、网络层安全、共识层安全、激励层安全、合约层安全、应用层安全）	理解
		常见的漏洞和攻击手段	理解
智能合约	概念与定义	智能合约的定义、相关概念	理解
	智能合约模型	智能合约的运行机制	理解
		智能合约的架构模型	理解
	智能合约开发	以太坊	掌握
		超级账本	掌握
	分布式网络	P2P 网络	理解

知识领域	知识单元	知识点	要求
分布式账本（区块链）系统环境构建与运行维护		节点发现	理解
	系统环境构建	搭建私有环境链	掌握
		Bash 脚本运用	掌握
		参数配置与调整	掌握
	区块链架构调整	节点维护	理解
“区块链+”实际场景应用分析	以太坊	区块数据检索	掌握
		区块数据插入	掌握
		应用开发	掌握
	超级账本	节点、排序节点、通道	掌握
		区块数据检索	掌握
		区块数据插入	掌握
		应用开发架构	理解
	典型场景分析	能源、金融、医疗等	理解
		具体案例解读	理解

备注：

1. 对知识和技能的考核要求中，二级为知道/理解/掌握，三级全部为掌握。

2. 知识与技能的学习考核要求分为**知道**、**理解**和**掌握**三个层次，其含义分别为：

知道：能识别和记忆相关的学习内容，对相关的知识有初步认识。

理解：初步把握学习内容的由来、作用和使用方法，并能以相应的学习内容为主完成简单的程序编制。

掌握：以某一学习内容为重点，综合运用其他相关内容，实现给定问题下的程序编制。

四、试卷结构

题号	题型	题量	分值	考核内容	考核目标
一	单选题	15 题	45 分	链式结构、系统能力、密码学基础、分布式网络、共识机制、智能合约、分布式账本（区块链）安全等基础知识	区块链基本概念 区块链架构与操作 智能合约开发
二	填空题	5 题	30 分	分布式账本（区块链）系统架构知识 环境构建与运维知识	区块链架构与操作 智能合约开发 场景分析能力
三	程序填空/改错题	2 题	30 分	依照要求开发可以满足场景需求的智能合约，在沙盒环境运行	智能合约开发
四	综合分析题	2-3 题	45 分	根据实际场景目标进行分析，对比分析适用于该场景的不同信息系统架构模	场景分析能力

题号	题型	题量	分值	考核内容	考核目标
				式优缺点，根据实际场景目标选择合适的分布式账本（区块链）架构，设计节点协作模式，选择合适的共识模型，提出业务流程改造中利用分布式账本（区块链）能力的分析过程	
合 计		22-23 题	150 分		

五、相关说明

1. 考试时间：150 分钟。
2. 试卷总分：满分 150 分。
3. 等第：不合格、二级合格、二级优秀、三级合格、三级优秀。各等第分数线由考委会划定。
4. 考试方式：考试采用基于网络环境的无纸化上机考试。
5. 考试环境：
   - 上海市高等学校信息技术水平考试通用平台。
   - 操作系统：Linux（以容器环境运行于 Windows）。
   - 程序运行环境：超级账本：2.0。
   - 智能合约：Solidity：^0.6.0。
   - 智能合约：GOLANG。
6. 建议学时数： 48-64 学时，其中实验课不少于 20 学时。
7. 参考教材：
   - 《分布式账本（区块链）基础与实践》（刘百祥、阚海斌等编著），复旦大学出版社，2020 年。

六、题型示例

单选题

【例】比特币区块间维持链式结构的方式（ ）

A.时间戳

B.默克尔树

C.区块高度

D.哈希指针

【参考答案】 D

【能力目标】理解不可篡改的能力

【知识内容】区块链链式结构，Hash 函数

填空题

【例】中国在央行数字货币研发中位于国际领先位置，人民银行推出的数字货币名为_____。

【参考答案】 DC/EP，DCEP

【能力目标】了解数字货币的常识

【知识内容】央行数字货币

程序填空题

【例】 下面的代码是使用 Go 语言所实现的超级账本的智能合约 MyContract 的部分代码，请完成链码被 peer 调用进行交易执行的接口。已知如下 API：

```
// GetArgs returns the arguments intended for the chaincode Init and Invoke
// as an array of byte arrays.
GetArgs() [][]byte

// GetStringArgs returns the arguments intended for the chaincode Init and
// Invoke as a string array. Only use GetStringArgs if the client passes
// arguments intended to be used as strings.
GetStringArgs() []string

// GetFunctionAndParameters returns the first argument as the function
// name and the rest of the arguments as parameters in a string array.
// Only use GetFunctionAndParameters if the client passes arguments intended
// to be used as strings.
```

```
        GetFunctionAndParameters() (string, []string)

    // GetState returns the value of the specified `key` from the
    // ledger. Note that GetState doesn't read data from the writeset, which
    // doesn't
    // consider data modified by PutState that has not been committed.
    // If the key does not exist in the state database, (nil, nil) is returned.
    GetState(key string) ([]byte, error)

    // PutState puts the specified `key` and `value` into the transaction's
    // writeset as a data-write proposal. PutState doesn't effect the ledger
    // until the transaction is validated and successfully committed.
    // Simple keys must not be an empty string and must not start with a
    // null character (0x00) in order to avoid range query collisions with
    // composite keys, which internally get prefixed with 0x00 as composite
    // key namespace. In addition, if using CouchDB, keys can only contain
    // valid UTF-8 strings and cannot begin with an underscore ("_").
    PutState(key string, value []byte) error

    // DelState records the specified `key` to be deleted in the writeset of
    // the transaction proposal. The `key` and its value will be deleted from
    // the ledger when the transaction is validated and successfully committed.
    DelState(key string) error
```

```
type MyContract struct {
}

// 实现链码在初始化和升级时调用的接口，初始化相关的数据。
func /*（1）将定义初始化函数的语句补充完整*/(stub  shim.ChaincodeStubInterface)
peer.Response {
```

```
    args := /*（2）以字符串方式获取调用该接口的参数列表*/
    if len(args) != 2 {
        return shim.Error("Incorrect arguments. Expecting a key and a value")
    }
    _, err := strconv.Atoi(args[1])
    if err != nil {
        return shim.Error("Expecting integer value for state")
    }
    //将参数中的Key/Value添加到账本中
    err = /*（3）向账本存储数据*/
        if err != nil {
        return shim.Error(fmt.Sprintf("Failed to create state: %s", args[0]))
    }
    return shim.Success(nil)
}

func (t *MyContract) get(stub shim.ChaincodeStubInterface, args []string)
peer.Response {
    if len(args) != 1 {
        return shim.Error("Incorrect arguments. Expecting a key ")
    }
    key := arg[0]
    valBytes, err := /*（4）从账本中获取键key的值*/
    if err != nil {
        return shim.Error(fmt.Sprintf("Fail to get state: %s", key))
    }
    if valBytes == nil {
        return shim.Error(fmt.Sprintf("Nil amount: %s.", key))
    }
```

```
    return shim.Success(valBytes)
}
```

【参考答案】

（1） (t *MyContract) Init

（2）stub.GetStringArgs()

（3）stub.PutState(args[0], []byte(args[1]))

（4）stub.GetState(key)

【能力目标】依照要求开发可以满足场景需求的智能合约，在沙盒环境运行

【知识内容】go 语言，智能合约 API

综合分析题

【例】请阅读如下材料，完成项目的区块链架构设计，在文件中填写完成所需步骤。

Hyperledge fabric 用来搭建联盟链，现有一个区块链+项目：基于区块链的长三角学分银行数据跨域流通平台项目，它的目标是将区块链技术应用于个人终生学习的学分信息数据、证书数据（两类数据）的跨区域认定、转换、迁移场景，可有效支撑长三角地区教育资源一体化共享利用，服务于人才资源跨区域流动。已知共有 4 个省的学分银行(A,B,C,D)参与学分银行数据共享，包括 2 个合约(CC1，CC2)，2 个省的学分银行(A,C)试点证书数据共享，包括 2 个合约(CC3，CC4)，请设计并补全操作流程。

【参考答案】

1.步骤一：生成_6_个节点，包括_4 个 peer，2 个 orderer_；

2.步骤二：生成__2__个消息通道，包括__Channel1，Channel2__；

3.步骤三：分别将节点和 orderer 加入两通道，ABCD 加入一个，AC 加入一个；

4.步骤四：分别在通道部署合约 CC1，CC2，CC3，CC4_。

【能力目标】根据实际场景目标进行分析，对比分析适用于该场景的不同信息系统架构模式优缺点，根据实际场景目标选择合适的分布式账本（区块链）架构，设计节点协作模式，选择合适的共识模型，提出业务流程改造中利用分布式账本（区块链）能力的分析过程。

【知识内容】典型场景分析