COSC 5010 - Blockchain Design and Programming Lecture Note 3

Libao Jin (ljin1@uwyo.edu)

December 11, 2018

1 Fall 2018 COSC Study Guide for Final Exam

1.1 Short Essay Questions will be from:

1. Explain a zero knowledge proof to a layman.

   Read the section "Ali Baba Cave" in Wikipedia on Zero Knowledge Proof. https://en.wikipedia.org/wiki/Zero-knowledge_proof. Be able to reproduce the cave diagram and write a short explanation of it.

2. Explain zk-SNARK - What is the "SNARK" part of zk-SNARK.

   "zk" is for zero knowledge. SNARK is (S, N, AR, K)

   - Succinct – the sizes of the hash functions (proofs) are tiny in comparison to the length of the actual process required to create them.
   - Non-interactive – there is no or only little interaction. For zk-SNARKs, there is usually a setup phase and after that a single point of contact between the prover and the verifier. Furthermore, SNARKs often have the so-called "public verifier" property meaning that anyone can verify without interacting anew, which is important for blockchains.
   - ARguments – the verifier is only protected against computationally limited provers. Provers with enough computational power can create proofs/arguments about wrong statements. (Note that with enough computational power, any public-key encryption can be broken). This is also called "computational soundness," as opposed to "perfect soundness."
   - of Knowledge – it is not possible for the prover to construct a proof/argument without knowing a certain so-called witness. (For example the address he wants to spend from, or the path to a certain Merkle-tree node).

3. How can blockchain change accounting?

   Double entry accounting was first recorded in 1340 in Genoa, Italy. It has remained basically the same since. We have automated the same system that has been used for hundreds of years. Blockchain provides a mathematically provable way of implementing accounting so that the most common form of accounting fraud is impossible to do. This is the first real "technological" advance in accounting in 500+ years.

4. When was double entry accounting invented?

   Double entry accounting was first recorded in 1340 in Genoa Italy.

5. What is a blockchain? - Why is blockchain important? - What are blockchains used for?

   A blockchain is a set of blocks that are written over time, and stored in a publicly accessible way. Each block is cryptographically signed and the signatures are used to chain the blocks together. The blocks are distributed so that every computer in the network has a copy of the blocks. Blockchain is the underlying technology behind BitCoin and other Cryptocurrencies.

6. Why is blockchain important?

   Blockchain creates trust between un-trusting parties. It is a system for verifying that data that is entered has not changed over time. It creates an immutable data store that can be verified mathematically.

7. What is a smart contract?

   A smart contract is a program that runs on a blockchain. It is run by the "miners" that run and maintain the chain.

8. What is "mining" in the context of a blockchain system?

   Mining is the process of transferring funds from account to account and running of smart contracts. It also implements the digital signatures, (hashes), that provide security for a blockchain.

9. What are the good properties of a blockchain?

   - Immutable data
   - Distributed data
   - Verifiable data
   - Distributed source of trust

10. What are the bad properties of a blockchain?

    - Slowest database around.
    - Most expensive database around.

11. How do blockchains create trust between non-trusting parties?

    The data is distributed and verifiable by all parties. Trust is no longer a part of an institution, it is a mathematically verifiable system.

12. Why is a blockchain an expensive database?

    Ethereum has 12,000 nodes, (computers), that have to all update the storage of any data. It is not cheap to update 12,000 computers every time.

13. Why are blockchains a so slow?

    Ethereum has 12,000 nodes, (computers). It takes time to update a large set of computers. Every computer has to see every block of data and append every verified block to the chain. Every new block of data has to be verified. Mining also takes time.

14. What is Ethereum "gas" and how does this relates to Turing-Complete?

    - Gas is basically the internal pricing for running a transaction or a contract. The gas price per transaction or contract is set up to deal with the Turing Complete nature of Ethereum and its EVM (Ethereum Virtual Machine Code)
    - Ethereum provides a decentralized virtual machine, the Ethereum Virtual Machine (EVM), which can execute scripts using an international network of public nodes. The virtual machine's instruction set, in contrast to others like Bitcoin Script, is expressive enough as languages like C that it's said to be Turing-complete in an informal sense. "Gas", an internal transaction pricing mechanism, is used to mitigate spam and allocate resources on the network.

15. Why is it difficult to store private data on a public chain?

16. Why is random data difficult to use on a blockchain?

1.2 Finance

1. What is the Yield Curve?

   A yield curve is a line that plots the interest rates, at a set point in time, of bonds having equal credit quality but differing maturity dates. The most frequently reported yield curve compares the three-month, two-year, five-year, 10-year and 30-year U.S. Treasury debt. This yield curve is used as a benchmark for other debt in the market, such as mortgage rates or bank lending rates, and it is used to predict changes in economic output and growth.

2. An inverted or down-sloped yield curve suggests yields on longer-term bonds may continue to fall, corresponding to periods of economic recession.

3. A Mutual Fund is an investment vehicle made up of a pool of money collected from many investors for the purpose of investing in securities such as stocks, bonds, money market instruments and other assets. A mutual fund's portfolio is structured and maintained to match the investment objectives stated in its prospectus.

1.3 Coding:

1. Solidity - a simple contract to store the hash of a document and a name for the document.

   - When is the contract constructor run?
   - Who can call the functions?
   - What do the 2 functions do?

```
pragma solidity ^0.4.24;
import "openzeppelin-solidity/contracts/ownership/Ownable.sol";

contract DocumentReg is Ownable {
    mapping(bytes32 => bool) infoSet;
    mapping(bytes32 => address) infoOwner;
    mapping(bytes32 => string) infoData;
    mapping(bytes32 => string) infoName;
    uint256 minPayment;
    event DocumentSet(string, bytes32, string);

    constructor() public {
        minPayment = 0;
    }

    function setPayment ( uint256 p ) public onlyOwner {
        minPayment = p;
    }

    function newInfo (  string name, bytes32 infoHash, string info ) public payable {
        require(!infoSet[infoHash], "already set, already has owner.");
        require(msg.value >= minPayment, "insufficient payment to set data.");
        infoSet[infoHash] = true;
        infoOwner[infoHash] = msg.sender;
        infoData[infoHash] = info;
        infoName[infoHash] = name;
        emit DocumentSet(name, infoHash, info);
    }

    function updateInfo (  bytes32 infoHash, string info ) public payable {
        require(infoOwner[infoHash] == msg.sender, "not correct owner of this data.");
        require(msg.value >= minPayment, "insufficient payment to update data.");
        string storage name;
        name = infoName[infoHash];
        infoData[infoHash] = info;
        emit DocumentSet(name, infoHash, info);
    }
}
```

2. Convert arithmetic to Go "big" for `b := 10; a = 12 * b + 31;`.

```
b = big.NewInt(10)
```

```
a = big.NewInt(0)
a.Add(big.NewInt(0).Mul(big.NewInt(12), b), big.NewInt(31))
```

```go
package main

import (
    "crypto/rand"
    "fmt"
    "math/big"
    "os"
)

// From: https://arrow.dit.ie/cgi/viewcontent.cgi?article=1031&context=itbj
// Id-in-crypto.pdf -- See p. 38, 40, 41 ((By the page numbers in the documetn))
// p40 (page number) is page 30 in my PDF viewer.  Look for section 7.9.

func main() {
    // From p40
    p := big.NewInt(88667)
    q := big.NewInt(1031)
    alpha := big.NewInt(70322)
    a := big.NewInt(755)

    // v = (alpha^(-a)) % p
    v := big.NewInt(0)
    v.Exp(alpha, big.NewInt(0).Neg(a), p)
    fmt.Printf("Setup Complete: v=%s\n", v)

    // Alice is the Client:

    // ------------------------------------------------------------
    // Message 1 - Client to Server
    // ------------------------------------------------------------

    // Alice Chooses, and send to Bob
    // r := big.NewInt(543) // Should be random, but for this example
    M := big.NewInt(9999) // Generate crypto random value
    r, err := rand.Int(rand.Reader, M)
    if err != nil {
        fmt.Printf("Failed to generate random value, %s\n", err)
        os.Exit(1)
    }

    fmt.Printf("r=%s\n", r)
    x := big.NewInt(0)
    x.Exp(alpha, r, p) // x=(alpha^r) % p
    fmt.Printf("Send To Bob : x=%s\n", x)

    // ------------------------------------------------------------------------
    // Response to Message 1, Server back to client
    // ------------------------------------------------------------------------

    // Bob is the Server:
    // Bob sends the challenge 'e' back to Alice e to do the computation
    // e = 1000
    // e := big.NewInt(1000)
    e, err := rand.Int(rand.Reader, M)
    if err != nil {
        fmt.Printf("Failed to generate random value, %s\n", err)
        os.Exit(1)
    }
    fmt.Printf("e=%s\n", e)

    // Alice now computes: y = r  +  a*e % q
    y := big.NewInt(0)
    y.Add(r, big.NewInt(0).Mod(big.NewInt(0).Mul(a, e), q))
    fmt.Printf("y=%s\n", y) // Prints 851
```

```
65
66      // ------------------------------------------------------------------------
67      // Message 2 - Client (Alice) with response to challenge.
68      // ------------------------------------------------------------------------
69
70      // Bob (server) verifies: x == z == (alpha^y) * (v^e) % p
71      // z = alpha^y * v^e % p
72      z := big.NewInt(1)
73      z.Mod(big.NewInt(0).Mul(big.NewInt(0).Exp(alpha, y, nil), big.NewInt(0).Exp(v, e, nil)), p)
74      fmt.Printf("z=%s\n", z)
75
76      // ------------------------------------------------------------------------
77      // Response 2 - Success/Fail message from server back to client
78      // ------------------------------------------------------------------------
79      if x.Cmp(z) == 0 {
80          fmt.Printf("Authoized! Yea\n")
81      } else {
82          fmt.Printf("Nope nope nope\n")
83      }
84  }
```

2 Fall 2018 COSC Study Guide for Midterm Exam

2.1 Terms with explanation

The test is on Monday, October 29. All You need to bring to class that day are a couple of pens. Standard Academic rules apply. The test will be 32 multiple choice/ fill in the blank. An example follows:

```
A corporation needs to raise money from the public it does this though
    A. initial public offering
    B. private offering
    C. going private
    D. fund raising from wealthy individuals
```

These are the terms you will need to study. The terms come from our in-class lectures and discussions, looking over your notes from class will be useful.
There will be 2 coding questions after the multiple choice section. Also there is 1 or 2 true false question.

1. Cash: Cash is also known as money, in physical form. Cash, in a corporate setting, usually includes bank accounts and marketable securities, such as government bonds and banker's acceptances. Although cash typically refers to money in hand, the term can also be used to indicate money in banking accounts, checks or any other form of currency that is easily accessible and can be quickly turned into physical cash.

   Cash in its physical form is the simplest, most broadly accepted and reliable form of payment, which is why many businesses only accept cash. Checks can bounce and credit cards can be declined, but cash in hand requires no extra processing. However, it's become less common for people to carry cash with them, due to the increasing dependability and convenience of electronic banking and payment systems.

   In finance and banking, cash indicates the company's current assets, or any assets that can be turned into cash within one year. A business's cash flow shows the net amount of cash a company has, after factoring in both incoming and outgoing cash and assets, and can be a good resource for potential investors. A company's cash flow statement shows all incoming cash, such as net income, and outgoing cash used to pay expenses such as equipment and investments.

2. Stock: You have probably heard a popular definition of what a stock is: "A stock is a share in the ownership of a company. Stock represents a claim on the company's assets and earnings. As you acquire more stock, your ownership stake in the company becomes greater." Unfortunately, this definition is incorrect in some key ways.

To start with, stock holders do not own corporations; they own shares issued by corporations. But corporations are a special type of organization because the law treats them as legal persons. In other words, corporations file taxes, can borrow, can own property, can be sued, etc. The idea that a corporation is a "person" means that the corporation owns its own assets. A corporate office full of chairs and tables belong to the corporation, and not to the shareholders.

This distinction is important because corporate property is legally separated from the property of shareholders, which limits the liability of both the corporation and the shareholder. If the corporation goes bankrupt, a judge may order all of its assets sold – but your personal assets are not at risk. The court cannot even force you to sell your shares, although the value of your shares will have fallen drastically. Likewise, if a major shareholder goes bankrupt, she cannot sell the company's assets to pay off her creditors.

What shareholders own are shares issued by the corporation; and the corporation owns the assets. So if you own 33% of the shares of a company, it is incorrect to assert that you own one-third of that company; it is instead correct to state that you own 100% of one-third of the company's shares. Shareholders cannot do as they please with a corporation or its assets. A shareholder can't walk out with a chair because the corporation owns that chair, not the shareholder. This is known as the "separation of ownership and control."

3. What is a P&L: The profit and loss statement is a financial statement that summarizes the revenues, costs and expenses incurred during a specified period, usually a fiscal quarter or year. P&L statement is synonymous with the income statement. These records provide information about a company's ability or inability to generate profit by increasing revenue, reducing costs or both. Some refer to the P&L statement as a statement of profit and loss, income statement, statement of operations, statement of financial results or income, earnings statement and expense statement.

   P&L statement is synonymous with the income statement. The income statement summarizes income and expenses. These records provide information about a company's ability or inability to generate profit by increasing revenue, reducing costs or both The balance sheet shows assets, liabilities, and owner's equity. The cash flow statement summarizes your incoming and outgoing money from operations, investing, and financing.

   Income statement vs. balance sheet: The income statement answers whether the business is profitable whereas the balance sheet shows what a company is owed and what it owns.

4. Long on Cash, Long in a Stock : this means that you have lots of cash, or that you own some of a stock. A long (or long position) is the buying of a security such as a stock, commodity or currency with the expectation that the asset will rise in value. In the context of options, long is the buying of an options contract. An investor that expects an asset's price to fall will go long on a put option, and an investor that hopes to benefit from an upward price movement will be long a call option.

   Very often you will read in a "disclosure" section that a person is "Long on …, …". This is to disclose that if they are saying good things about a company that they have a self-interest in you purchasing shares in the company. For a person writing professionally this is a legal requirement.r

5. "Take a position in"… The state of owning or owing a security or other asset. One has a long position when one owns something, while one has a short position when something is sold, especially sold short.

6. Real Estate: Land and the improvements on it. Real estate is one of the primary (and indeed one of the only) assets whose value does not depreciate over time. Depending on the particular title, ownership of real estate may include mineral rights to any geophysical aspects occurring thereon. Ownership of real estate does not automatically include the right to develop it, depending on local regulations. However, development of real estate (for example by building a house on it) usually increases the value. While supply of real estate does not vary, demand may change greatly depending on its particular features, number of people in the area, and cultural differences regarding land ownership. It is an attractive form of collateral because it cannot be stolen or destroyed.

7. What is Inflation? Inflation is the rate at which the general level of prices for goods and services is rising and, consequently, the purchasing power of currency is falling. Central banks attempt to limit inflation — and avoid deflation — in order to keep the economy running smoothly.

8. REIT (Real Estate Investment Trust) - legally required to pay out 80% of profits to investors. A real estate investment trust, or REIT, is a company that owns, operates or finances income-producing real estate. For a company to qualify as a REIT, it must meet certain regulatory guidelines. REITs often trades on major exchanges like other securities and provide investors with a liquid stake in real estate.

9. Price to Earnings Radio - In essence, the price-earnings ratio indicates the dollar amount an investor can expect to invest in a company in order to receive one dollar of that company's earnings. This is why the P/E is sometimes referred to as the price multiple because it shows how much investors are willing to pay per dollar of earnings.

   There will be a question on how to calculate a PE ratio on the test. The formula for calculating it will be on the test.

10. What is a Derivative? In finance, a derivative is a contract that derives its value from the performance of an underlying entity. This underlying entity can be an asset, index, or interest rate, and is often simply called the "underlying."

11. High speed trading: High-frequency trading - HFT is a program trading platform that uses powerful computers to transact a large number of orders at fractions of a second. It uses complex algorithms to analyze multiple markets and execute orders based on market conditions. Typically, the traders with the fastest execution speeds are more profitable than traders with slower execution speeds.

12. Dividends: A dividend is a distribution of a portion of a company's earnings, decided by the board of directors, paid to a class of its shareholders. Dividends can be issued as cash payments, as shares of stock, or other property.

13. What is the "Yield"? In finance, the yield on a security is the amount of cash (in percentage terms) that returns to the owners of the security, in the form of interest or dividends received from it. Normally, it does not include the price variations, distinguishing it from the total return.

    There will be a question on calculating the Yield from the stock price and the amount of dividends on the test.

14. What is a basis point (BPS)? Example 50 BSP = 0.5%: Basis points, otherwise known as bps or "bips," are a unit of measure used in finance to describe the percentage change in the value or rate of a financial instrument. One basis point is equivalent to 0.01% (1/100th of a percent) or 0.0001 in decimal form.

    There will be a question of converting 75 BSP to a percentage on the test.

15. Stock Buy Back: A stock buyback, or repurchase, occurs when a company buys its own shares off the market and therefore reduces the amount of stock outstanding. It can do this in one of two ways: The company can either buy shares at current market prices or tender a fixed-price offer to current shareholders.

    Stock buy backs are a good indication that the company believes that its stock is a good investment. Also this is a way for the company to directly influence the price of the stock. Fewer shares sell for more for each share. This also will increase the dividends that are payd out.

    Stock buy backs are a bad thing when the company has to borrow to pay for them.

16. Index Fund: An index fund is a type of mutual fund with a portfolio constructed to match or track the components of a market index, such as the Standard & Poor's 500 Index (S&P 500). An index mutual fund is said to provide broad market exposure, low operating expenses and low portfolio turnover.

17. Insider Trading: Insider trading is the trading of a public company's stock or other securities (such as bonds or stock options) by individuals with access to nonpublic information about the company. In various countries, some kinds of trading based on insider information is illegal.

18. Bonds: A bond is a fixed income investment in which an investor loans money to an entity (typically corporate or governmental) which borrows the funds for a defined period of time at a variable or fixed interest rate. … Owners of bonds are debtholders, or creditors, of the issuer.

19. ICOs (Initial Coin Offering): An initial coin offering or initial currency offering is a type of funding using cryptocurrencies. Mostly the process is done by crowdfunding but private ICO's are becoming more common.

20. "Industrial Staker Sale" a token sale under a different name that isn't ICO.

21. Proof-of-Work describes a system that requires a not-insignificant but feasible amount of effort in order to deter frivolous or malicious uses of computing power, such as sending spam emails or launching denial of service attacks. In blockchain a proof of work is a race that is payed for to get a reward. This is how Bitcoin has been distributed.

22. Proof-of-Stake concept states that a person can mine or validate block transactions according to how many coins he or she holds. This means that the more Bitcoin or altcoin owned by a miner, the more mining power he or she has.

    The "stake" is at risk when mining occurs. If the miner lies then it looses some or all of its stake.

23. "Consumer Token Sale": Token sales are, quite simply, a process of generating and selling a new cryptocurrency. While the details change from sale to sale, this process involves building a smart contract on the blockchain, generating, and then selling the resulting coins.

24. Junk Bonds refers to high-yield or noninvestment-grade bonds. Junk bonds are fixed-income instruments that carry a credit rating of BB or lower by Standard & Poor's, or Ba or below by Moody's Investors Service. Junk bonds are so called because of their higher default risk in relation to investment-grade bonds.

25. Mutual Fund is an investment vehicle made up of a pool of money collected from many investors for the purpose of investing in securities such as stocks, bonds, money market instruments and other assets. … A mutual fund's portfolio is structured and maintained to match the investment objectives stated in its prospectus.

    Mutual funds have both a managed and an index variety. Usually the costs of a managed mutual fund and the tax that you pay on it will make it less profitable than an index fund. Consider the returns on the Vanguard S&P 500 index fund over the last 25 years. Only 1% of managed mutual funds have out performed the Vanguard S&P 500 Fund.

26. Asset allocation refers to the strategy of dividing your investments among different asset categories, such as stocks, bonds, real estate, cash, and cash alternatives. Asset allocation aims to control risk by diversifying an investment portfolio.

27. Expense Ratio for a Mutual Fund. The expense ratio, also known as the management expense ratio (MER), measures how much of a fund's assets are used for administrative and other operating expenses. An expense ratio is determined by dividing a fund's operating expenses by the average dollar value of its assets under management (AUM).

28. Prospectus. A prospectus, in finance, is a disclosure document that describes a financial security for potential buyers. … In the context of an individual securities offering, such as an initial public offering, a prospectus is distributed by underwriters or brokerages to potential investors.

29. Pro Forma: Pro forma is a Latin term that describes a method of calculating and presenting financial results to emphasize either current or projected figures. Pro forma literally means "for the sake of form" or "as a matter of form." In the world of investing, pro forma refers to a method by which firms calculate financial results. This method of calculation emphasizes present or projected figures.

30. KYI - Know Your Investor (See SEC 506(d)).

    You can have up to 34 friends and family as an exemption.

31. Accredited Investor. An accredited investor is a person or entity that can deal with securities not registered with financial authorities by satisfying one of the requirements regarding income, net worth, asset size, governance status or professional experience. The term is used by the Securities and Exchange Commission (SEC) under Regulation D to refer to investors who are financially sophisticated and have a reduced need for the protection provided by regulatory disclosure filings. Accredited investors include natural individuals, banks, insurance companies, brokers and trusts.

32. Going Public. Going public refers to a private company's initial public offering (IPO), thus becoming a publicly traded and owned entity. Businesses usually go public to raise capital in hopes of expanding. Venture capitalists may use IPOs as an exit strategy (a way of getting out of their investment in a company).

33. Money Laundering. Money laundering is the process of creating the appearance that large amounts of money obtained from criminal activity, such as drug trafficking or terrorist activity, originated from a legitimate source. The money from the illicit activity is considered dirty, and the process "launders" the money to make it look clean.

34. Cost of "going public." Going public is the process of selling shares that were formerly privately held to new investors for the first time. Otherwise known as an initial public offering (IPO). Going public is very expensive. Usually a company today needs $250,000,000.00 in revinue to justify going public. Before Enron it was common to see companies go public with a revinue of $10,000,000.00 a year.

2.2 Go Code for Test

1. Declare structure
2. Serialize - A structure: `tx/transaction.go`, `block/block.go`, `s := fmt.Sprintf("%10d: %s", iv, s)`
3. Hash - Call a hash function: `h := Keccak256([]byte)`
4. JSON - Read / Write
5. Map - Declare, Print
6. Declare function - Return values

- `tutorial.go`

```go
package main

import (
    "fmt"
    "math"
)

// import ("fmt"; "math")

func main() {
    fmt.Println(`String`[2:5])
    fmt.Println(`abcd \n dsfds`)
    fmt.Printf("%c\n", `String`[0])
    fmt.Println(len(`String`))
    fmt.Println(true)
    fmt.Println(false)
    fmt.Println(true && true)
    fmt.Println(false && true)
    fmt.Println(false || true)
    fmt.Println(321325 * 424521)
    var s string = "Hey, this is a variable storing a string."
    fmt.Println(s)
    var x string = "String 1"
    var y string = " String 2"
    fmt.Println(x + y)
    fmt.Println(x == y)
    x += y
    fmt.Println(x)
```

```
29      var z = "Hey"
30      fmt.Println(z)
31      a := "???"
32      fmt.Println(a)
33      dogsName := "Max"
34      fmt.Println("My dog's name is", dogsName)
35      fmt.Println("My dog's name is", dogsName, "Great")
36      const cs1 string = "Hello world 1"
37      const cs2 = "Hello world 2"
38      fmt.Println(cs1, cs2)
39      fmt.Println(math.Pi)
40      var (
41          e = 5
42          f = 4
43          g = 10
44      )
45      fmt.Println(e + f + g)
46      fmt.Println("Enter an floating point number: ")
47      var input float64
48      fmt.Scanf("%f", &input)
49      var output float64 = 2 * input
50      fmt.Println(output)
51      i := 0
52      for i < 10 {
53          fmt.Println(i)
54          i += 1
55      }
56
57      for i := 0; i < 10; i++ {
58          if i%2 == 0 {
59              fmt.Println(i, "even")
60          } else {
61              fmt.Println(i, "odd")
62          }
63      }
64
65      for i := 0; i < 5; i++ {
66          switch i {
67          case 0:
68              fmt.Println(i, "Zero")
69          case 1:
70              fmt.Println(i, "One")
71          case 2:
72              fmt.Println(i, "Two")
73          default:
74              fmt.Println(i, "Not a recognizable number ")
75          }
76      }
77
78      // array
79      var arr [10]int
80      arr[4] = 5431
81      for i := 0; i < 10; i++ {
82          fmt.Println(arr[i])
83      }
84
85      var fruits [5]float64
86      fruits[0] = 1234.21
87      fruits[1] = 34.21
88      fruits[2] = 134.21
89      fruits[3] = 12.21
90      fruits[4] = 121
91
92      var total1 float64 = 0
93      for i := 0; i < 5; i++ {
94          total1 += fruits[i]
95      }
96      fmt.Println(total1 / 5)
```

```
97
98      var total2 float64 = 0
99      for i := 0; i < len(fruits); i++ {
100         total2 += fruits[i]
101     }
102     fmt.Println(total2 / float64(len(fruits)))
103
104     var total3 float64 = 0
105     for i, value := range fruits {
106         fmt.Println(i, value)
107         total3 += value
108     }
109     fmt.Println(total3 / float64(len(fruits)))
110
111     var total4 float64 = 0
112     for _, value := range fruits {
113         total4 += value
114     }
115     fmt.Println(total3 / float64(len(fruits)))
116
117     yy := [5]float64{63, 4312, 345, 543, 432}
118     for i, value := range yy {
119         fmt.Println(i, value)
120     }
121
122     for k, v := range "□ abd" {
123         // fmt.Println(k, v)
124         fmt.Printf("%d, %c, %T\n", k, v, v)
125     }
126
127     zz := [5]float64{
128         63,
129         4312,
130         345,
131         543,
132         432,
133     }
134     for i, value := range zz {
135         fmt.Println(i, value)
136     }
137
138     // fmt.Println("Golang is fun,
139     // ")
140     fmt.Println(`Golang is fun,
141     `)
142     fmt.Println("Formatting is tedious")
143
144     // slice
145     xxx := make([]float64, 5, 10)
146     for i, value := range xxx {
147         fmt.Println(i, value)
148     }
149
150     zzz := zz[2:4]
151     for i, value := range zzz {
152         fmt.Println(i, value)
153     }
154
155     var slice []int
156     fmt.Println(slice)
157
158     slice1 := []int{1, 2, 3, 4, 5}
159     slice2 := append(slice1, 6, 7, 8, 9)
160     slice3 := make([]int, 2)
161     slice4 := make([]int, 3, 9)
162     slice5 := slice2[3:7]
163     copy(slice3, slice2)
164     copy(slice4, slice2)
```

```
165     fmt.Println(slice1, slice2, slice3, slice4, slice5)
166     fmt.Println(len(slice4))
167     fmt.Printf("%T\t%v\t%d\n", slice4, slice4, slice4[0])
168
169     // map
170     // var mmm map[string]int
171     // mmm["key"] = 10
172     // mmm["answer"] = 32
173     // fmt.Println(mmm)
174     mmm := make(map[string]int)
175     mmm["key"] = 10
176     mmm["answer"] = 32
177     fmt.Println(mmm)
178     delete(mmm, "answer")
179     fmt.Println(mmm)
180     k, v := mmm["key"]
181     fmt.Println(k, v)
182     if k, v := mmm["key"]; v {
183         fmt.Println("Successful!", k)
184     }
185
186     elements := map[string]string{
187         "Fe": "Iron",
188         "H":  "Hydrogen",
189         "C":  "Carbon",
190     }
191     fmt.Println(elements)
192
193     fmt.Println(f1())
194     fmt.Println(f2())
195     fmt.Println(f3())
196     fmt.Println(sum(f3()))
197     fmt.Println(add(1, 2, 3, 4, 5))
198     xs := []int{1, 2, 3}
199     fmt.Println(add(xs...))
200
201     // closure
202     addition := func(x, y int) int {
203         return x + y
204     }
205     fmt.Println(addition(1, 1))
206
207     local_x := 0
208     increment := func() int {
209         local_x++
210         return local_x
211     }
212     fmt.Println(increment())
213     fmt.Println(increment())
214
215     nextEven := makeEvenGenerator()
216     fmt.Println(makeEvenGenerator())
217     fmt.Println(makeEvenGenerator())
218     fmt.Println(makeEvenGenerator()())
219     fmt.Println(makeEvenGenerator()())
220     fmt.Println(nextEven())
221     fmt.Println(nextEven())
222
223     defer third()
224     first()
225     second()
226     // f, _ := os.Open(filename)
227     // defer f.Close()
228
229     // panic("PANIC")
230     // str := recover()
231     // fmt.Println(str)
232
```

```
233     // var c Circle
234     // c := new(Circle)
235     // c := Circle{x: 0, y: 0, r: 5}
236     c := Circle{0, 0, 5}
237     fmt.Println(c.x, c.y, c.r)
238     fmt.Println(circleArea(c))
239     fmt.Println(c.area())
240     fmt.Println(circleCircumference2(&c))
241     fmt.Println(circleCircumference(&c))
242 
243     r := Rectangle{0, 0, 1, 1}
244     fmt.Println(r.area())
245 
246     fmt.Println(totalArea(&c, &r))
247 
248     android := new(Android)
249     android.Person.Name = "Josh"
250     android.Person.Talk()
251     // and.Talk()
252 
253     ios := new(iOS)
254     ios.Name = "iOS"
255     ios.Talk()
256 
257     defer func() {
258         str := recover()
259         fmt.Println(str)
260     }()
261     panic("PANIC")
262 }
263 
264 func f1() (r int) {
265     r = 1
266     return
267 }
268 
269 func f2() int {
270     r := 1
271     return r
272 }
273 
274 func f3() (int, int) {
275     return 5, 6
276 }
277 
278 func sum(a int, b int) int {
279     return a + b
280 }
281 
282 // variadic functions
283 func add(args ...int) int {
284     total := 0
285     for _, v := range args {
286         total += v
287     }
288     return total
289 }
290 
291 func makeEvenGenerator() func() uint {
292     i := uint(0)
293     return func() (ret uint) {
294         ret = i
295         i += 2
296         return
297     }
298 }
299 
300 func first() {
```

```
301     fmt.Println("1st")
302 }
303
304 func second() {
305     fmt.Println("2nd")
306 }
307
308 func third() {
309     fmt.Println("3rd")
310 }
311
312 type Circle struct {
313     x float64
314     y float64
315     r float64
316     // x, y, r float64
317 }
318
319 func circleArea(c Circle) float64 {
320     return math.Pi * c.r * c.r
321 }
322
323 func circleCircumference(c *Circle) float64 {
324     return math.Pi * c.r * 2
325 }
326
327 func circleCircumference2(c *Circle) float64 {
328     perimeter := math.Pi * (*c).r * 2
329     c.r = 10
330     return perimeter
331 }
332
333 func (c *Circle) area() float64 {
334     return math.Pi * c.r * c.r
335 }
336
337 type Rectangle struct {
338     x1, y1, x2, y2 float64
339 }
340
341 func (r *Rectangle) area() float64 {
342     return math.Abs(r.x1-r.x2) * math.Abs(r.y1-r.y2)
343 }
344
345 type Shape interface {
346     area() float64
347 }
348
349 func totalArea(shapes ...Shape) float64 {
350     var area float64
351     for _, s := range shapes {
352         area += s.area()
353     }
354     return area
355 }
356
357 type MultiShape struct {
358     shapes []Shape
359 }
360
361 func (m *MultiShape) area() float64 {
362     var area float64
363     for _, s := range m.shapes {
364         area += s.area()
365     }
366     return area
367 }
368
```

```
type Person struct {
    Name string
}

func (p *Person) Talk() {
    fmt.Println("Hi, my name is", p.Name)
}

type Android struct {
    Person Person
    Model  string
}

type iOS struct {
    Person
    Model string
}
```

- output.txt

```
rin
abcd \n dsfds
S
6
true
false
true
false
true
136409210325
Hey, this is a variable storing a string.
String 1 String 2
false
String 1 String 2
Hey
???
My dog's name is Max
My dog's name is Max Great
Hello world 1 Hello world 2
3.141592653589793
19
Enter an floating point number:
4.68
0
1
2
3
4
5
6
7
8
9
0 even
1 odd
2 even
3 odd
4 even
5 odd
6 even
7 odd
8 even
9 odd
0 Zero
1 One
2 Two
3 Not a recognizable number
4 Not a recognizable number
0
0
0
0
5431
0
0
```

```
0
0
0
307.168
307.168
0 1234.21
1 34.21
2 134.21
3 12.21
4 121
307.168
307.168
0 63
1 4312
2 345
3 543
4 432
0, a, int32
1, b, int32
2,  , int32
5, d, int32
0 63
1 4312
2 345
3 543
4 432
Golang is fun,

Formatting is tedious
0 0
1 0
2 0
3 0
4 0
0 345
1 543
[]
[1 2 3 4 5] [1 2 3 4 5 6 7 8 9] [1 2] [1 2 3] [4 5 6 7]
3
[]int   [1 2 3] 1
map[key:10 answer:32]
map[key:10]
10 true
Successful! 10
map[Fe:Iron H:Hydrogen C:Carbon]
1
1
5 6
11
15
6
2
1
2
0x10a2a00
0x10a2a00
0
0
0
2
1st
2nd
0 0 5
78.53981633974483
78.53981633974483
31.41592653589793
62.83185307179586
1
315.1592653589793
Hi, my name is Josh
Hi, my name is iOS
PANIC
3rd
```