PA2

Computer Programming for Engineers

Instructor: Younghoon Kim

Introduction

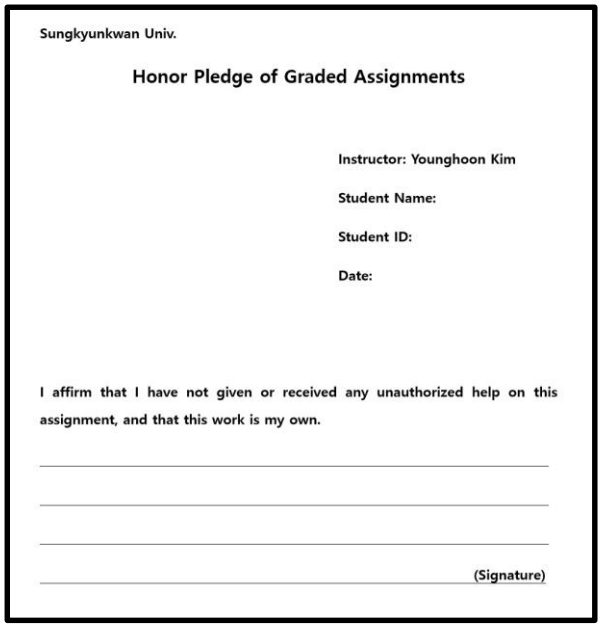
- **Check PA2 at http://skku.goorm.io/.**

- **Deadline : 2020.11.25 11:59 pm**
- **You can submit PA2 for two more days (~ 2020.11.27 11:59 pm) after the deadline. (25% deduction per day)**

- **Three problems for PP**
  - Each problem has the same portion.

Honor Pledge

- **Please upload the honor pledge to iCampus.**
  - The template of honor pledge will be uploaded on iCampus.
  - Download and print it. (Or you can use your tablet PCs.)
  - Fill in your personal information.
  - **Handwrite** the statement and leave your signature.
  - With the signature of yours, you agree with the disciplinary actions which will be followed after the honor pledge violation.



Sungkyunkwan Univ.

Honor Pledge of Graded Assignments

Instructor: Younghoon Kim

Student Name:

Student ID:

Date:

I affirm that I have not given or received any unauthorized help on this assignment, and that this work is my own.

(Signature)

List of Problems

1. Bank/Stock Account Management
2. Coder and Decoder
3. Shape

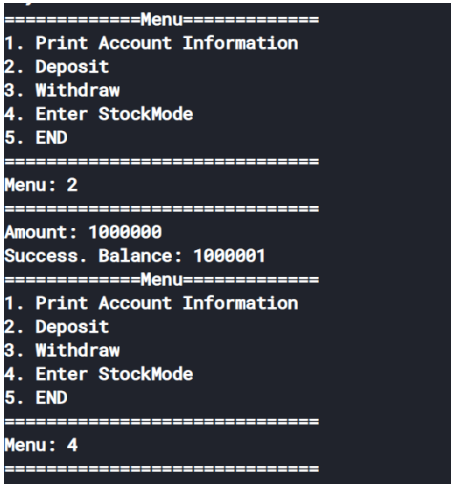
Problem 1. Bank/Stock Account Management

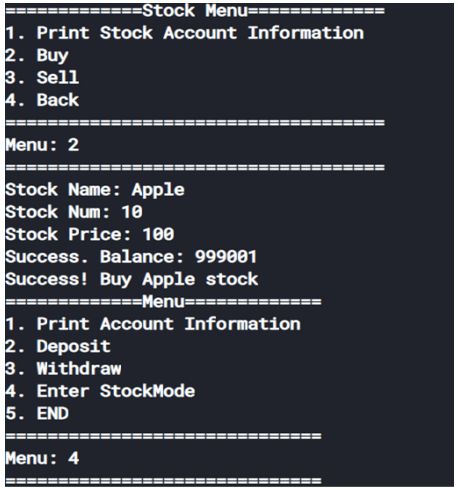
<Description>

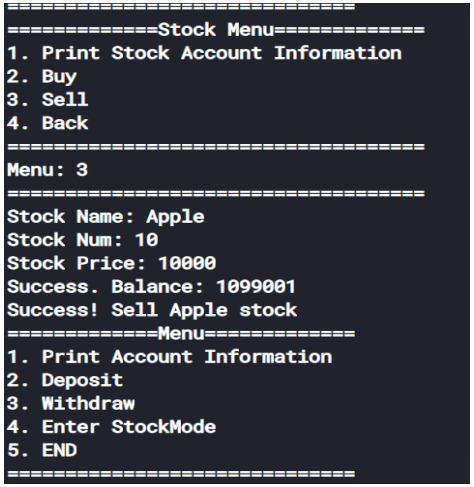
- There are two types of accounts: bank account and stock account.
- The stock account inherits the bank account.
- Functions of each account type
  - The bank account: deposit, withdraw
  - The stock account: deposit & withdraw (inherits the bank acc.) + stock buy & sell
- Suppose that there are only four stocks (Apple, Google, Samsung, Amazon)
- User selects a menu and performs functions.

<Score Policy> 33 scores per test case. (Total Score = 100)

<Output Example>







Problem 1. Bank/Stock Account Management

<Class Variables & Methods>

- name, age, balance: account information
- stock_nums: stock holdings
- printAccInfo(): shows each account information
  - Bank account: name, age, balance
  - Stock account: name, age, balance and stock info. (stock name & holdings)
- deposit(), withdraw(): increase/decrease the balance
- buyStock(): If there is an enough balance in the account (calls withdraw() and success), buy stocks.
- sellStock(): sell stock and deposit proceeds into the account.

```
#define STOCK_NUM 4
string stock_list[STOCK_NUM] = {"Apple", "Google", "Samsung", "Amazon"};
```

```
class BankAccount {
private:
  string name;
  int age;
  int balance;
public:
  BankAccount(string n, int a);
  void printAccInfo();
  void deposit(int amount);
  bool withdraw(int amount);
};
```

```
class StockAccount : public BankAccount {
private:
  int stock_nums[STOCK_NUM];

public:
  StockAccount(string n, int a);
  void buyStock(string stock_name, int num, int price);
  void sellStock(string stock_naume, int num, int price);
  void printAccInfo();
};
```

Problem 2. Coder and Decoder

<Description>

- Implement class Encoder and class Decoder.
  - Both classes inherit from class Coder

- Class Coder provides standard interface to both classes.
  - One virtual function named 'string translate(string input)'.
  - translate function will encode or decode some strings.

- class Encoder inherits Coder and override translate method.
  - translate function in Encoder class converts string to hexa decimal ASCII codes.
  - e.g. "abc" => "616263" for hexadecimal of 'a' will be 0x61, 'b' will be 0x62

- class Decoder also inherits Coder and override translate method.
  - translate function in Decoder class converts hexadecimal ASCII code to string.
  - e.g."595a5b" => "YZ[" for hexadecimal of 'Y' will be 0x59, 'Z' will be 0x5a
  - Please be aware of 0x5a is using small 'a' not capital ‘A’

Problem 2. Coder and Decoder

<Restrictions>

- Input words are limited to use only ASCII characters without special characters.
  - ABCabc!@#123[]{}-=,./<>? -> valid characters for inputs
  - \t\n\b -> Not used for inputs
- Hexadecimal must be small letters only.
  - 1a2b3c -> valid
  - 1A2B3C -> invalid

<Score Policy>

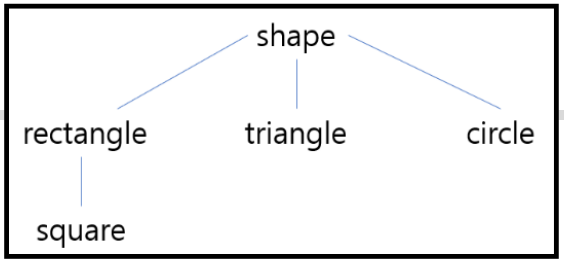
- 20 scores per test case. (Total Score = 100)

<Output Example>

```
Please enter a word: ABC
Encoded: 414243
Decoded: ABC
```

```
Please enter a word: YZ[
Encoded: 595a5b
Decoded: YZ[
```

Problem 3. Shape (1 / 2)



<Description>

- 'Circle', 'Triangle' and 'Rectangle' all have a common property of 'Shape', and 'Square' belongs to 'Rectangle'. This relationship is illustrated in the above figure.
- However, 'Circle', 'Triangle', 'Rectangle', and 'square' differ in how they calculate their area.
- Area of Circle, given radius r : $(3.141592) * r^2$
- Area of Triangle, given base b, height h : $(0.5) * b * h$
- Area of Rectangle, given width w, height h: $w * h$
- Area of Square, given side s : s^2

- Complete the code so that the output looks like the example screen.
- Input : type of the shape(string), information for calculating the area of the shape [width and height for rectangle, side for square, radius for circle, base and height for triangle]
- Output: the area of given shape

***** You must use inheritance of classes. if not, you cannot get a score**

Problem 3. Shape (2 / 2)

<Scoring Criteria>

- 20 scores per test case. (Total Score = 100)

<Console Output>

```
Input
rectangle
4 7
```

```
Output
Area of rectangle : 28
```

```
Input
square
7
```

```
Output
Area of square : 49
```