Project #1: Money Management

Develop a personal finance application that helps you manage your money wisely.

The application will ask you to enter an amount you earn in this month (Account #0). This income is divided into 6 sub-accounts which are described below:

- Account #1: **NEC** - Necessity - 55% of Account #0
- Account #2: **FFA** - Finance Freedom Account - 10% of Account #0
- Account #3: **EDU** - Education - 10% of Account #0
- Account #4: **LTSS** -Long Term Spending Saving - 10% of Account #0
- Account #5: **PLAY** - 10% of Account #0
- Account #6: **GIVE** - 5% of Account #0

Here’s an example:

```
Enter your income this month: 5000
```

Here’s how you should manage your money:

```
NEC: 2750          LTSS: 500

FFA: 500           PLAY: 500

EDU: 500           GIVE: 250
```

NOTE: Design the money dividing module in a way that is reusable in other applications, e.g. desktop, web and mobile.

Project #2: Human Resource Management

Develop a simple Human Resources Management (HRM) application for an IT company. The application manages the following kinds of employee:

- Programmer
- Designer
- Tester
- Accountant
- Business Developer
- Architect
- Team Leader
- Chairman
- CEO
- CTO
- CFO

You are required to develop the employee module which should be re-usable and extensible. Here’s the required information for this module:

- Employee is the super type of all kinds of employees in the organization.
    - Attributes: name, email, address, birthday
    - Behaviors: work
- Programmer is a kind of employee:
    - Attributes: programming languages
    - Behaviors: code, fix bugs
- Tester is a kind of employee:
    - Attributes: testing methodologies.
    - Behaviors: test, verify.
- Designer is a kind of employee:
    - Attributes: design philosophy, design tools used
    - Behavior: design
- Business developer is a kind of employee:
    - Attributes: specialized domains
    - Behaviors: collect requirements, analyze requirements, write documents.
- Team leader is a kind of programmer:
    - Attributes: team collaboration tools, development methods
    - Behaviors: schedule, organize, manage programmers, designers and testers.
- Architect is a kind of programmer:
    - Attributes: design methodology
    - Behavior: design system
- CTO is a kind of architect who has:
    - Attributes: technology domains
    - Behaviors: advise technologies
- CEO is a kind of employee who has:

+ Attributes: vision, mission

+ Behavior: executive management

- CFO is a kind of employee who has:

  + Attributes: finance management skills

  + Behaviors: manage finance

You are required to implement entity classes for all these kinds of people using OOP features abstraction, encapsulation and inheritance. Finally, package the compiled classes as a JAR file (employee.jar) so it can be re-used among applications.

Project #3: Numbers to Words

Develop a program that is able to read an arbitrary integer number from the command line, then prints out that number in words. For example:

- **Input**: `20` **Output**: `twenty`

- **Input**: `86` **Output**: `eighty six`

- **Input**: `365` **Output**: `three hundred and sixty five`

The program can read numbers up to 9999. Design this program in a way that can be re-used in other projects.

Project #4: Fibonacci Sequence

Write a program that prints the Fibonacci sequence numbers less than a given number N. For example:

- **Input**: `N = 10` **Output**: `1   1   2   3   5   8`

- **Input**: `N = 30` **Output**: `1   1   2   3   5  8   13   21`

- **Input**: `N = 60` **Output**: `1   1   2   3   5  8   13   21   34   55`

Project #5: Case Changer

Write a program that can change a given input string to the following types of case:

- All upper case
- All lower case
- Capitalize (capitalize the first letter of each word)
- Sentence case (capitalize the first letter of the first word in a sentence)
- Invert: lower to upper and upper to lower

Here’s an example:

- **Input**:

```
Mary has a little dog. she calls him Pun
```

- **Output:**

Upper case:

```
MARY HAS A LITTLE DOG. SHE CALLS HIM PUN
```

Lower case:

```
mary has a little dog. she calls him pun
```

Capitalize:

```
Mary Has A Little Dog. She Calls Him Pun
```

Sentence case:

```
Mary has a little dog. She calls him pun
```

Invert:

```
mARY HAS A LITTLE DOG. SHE CALLS HIM pUN
```

Project #6: Factorial Number

Write a program that calculates factorial of a given number N. It prints the output by both number and words (reuse the stuffs developed in the project #13).

For example:

- **Input:**

```
N = 5
```

- **Output**:

```
5! = 120 (one hundred twenty two)
```

Project #7: Triangle Area Calculator

Write a program that calculates area of a triangle, given its 3 points (A, B and C) in the Descartes coordinate system.

- Input:

```
Point A (x, y):    10 20

Point B (x, y):    10 60

Point C (x, y):    80 20
```

- Output:

```
Area = 1400
```

Project #8: Cards Dealer

Write a program that mimics a dealer in cards game. In each game, the dealer divides 52 cards for 4 players randomly. Suppose the following letters denote the suits:

- C: clubs - D: Diamond - H: Heart - S: Spades Here's an

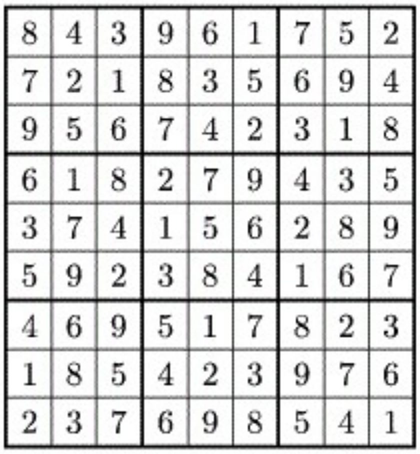
example:

```
Player #1: 3c  8d  2h  As  Jc  10h  Kh  5s  6s  9d  Qd  2s  7c

Player #2: 6d  6h  2d  3s  4d  4h  5c  Qc  Kd  Jd  8c  10c  Ac

Player #3: 3d  4c  Jh  6c  10d  5h  7s  2c  7h  Ad  8s  Qh 3h

Player #4: Ah  4s  10s  5d  Qs  Kc  Ks  9c  9h  9s  Js  8h  7d
```

Project #9: Sudoku Matrix

Write a program that randomly prints a Sudoku board (9x9 matrix) which is divided into 3 sub-matrices (3x3 each). Each sub matrix contains numbers ranging from 1 to 9.

Fill numbers into the whole board in ways so that there is no duplicate numbers on each row, each column and each 3x3 block. Here’s an example:



8	4	3	9	6	1	7	5	2
7	2	1	8	3	5	6	9	4
9	5	6	7	4	2	3	1	8
6	1	8	2	7	9	4	3	5
3	7	4	1	5	6	2	8	9
5	9	2	3	8	4	1	6	7
4	6	9	5	1	7	8	2	3
1	8	5	4	2	3	9	7	6
2	3	7	6	9	8	5	4	1

Project #10: Permutation

Write a program that prints permutations of any given set of numbers. For example:

Input:

```
Total number N  = 3

Number #1: 2

Number #2: 8

Number #3: 9
```

Output:

```
The permutations of (2, 8, 9) are: 6
(2, 8, 9)          (2, 9, 8)          (8, 2, 9)

(8, 9, 2)          (9, 2, 8)          (9, 8, 2)
```