# Comp151 Lab01

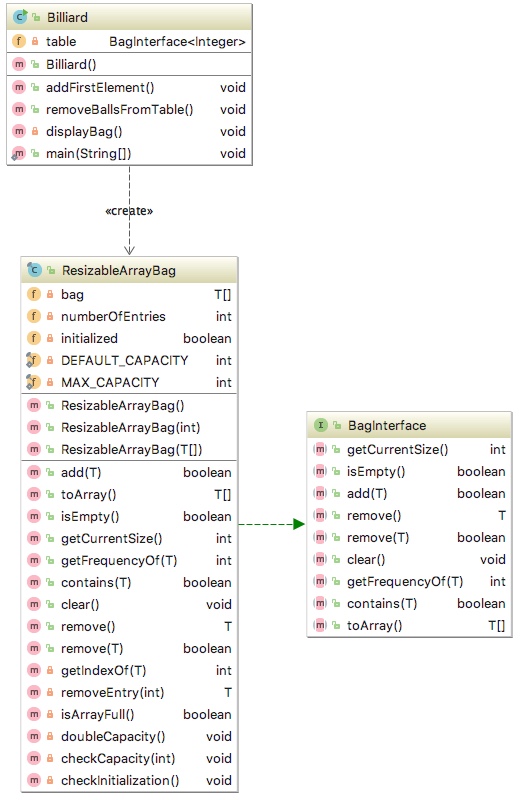
Write **three** independent applications as per descriptions below:

**Project1**

Suppose that you have several numbered billiard balls on a pool table. The smallest possible number on the ball is “1”. At each step, you remove a billiard ball from the table. If the ball removed is numbered *n*, you replace it with *n* balls randomly numbered less than *n*. For example, if you remove the “5” ball, you replace it with balls numbered “2”, “1”, “1”, “4”, and “3”, where numbers 2, 1, 1, 4, and 3 were randomly generated. If you remove the “1” ball, no new balls will be added.

Write a program that simulates this process. Start with only one ball on the table with the number on it selected by the user. Use the class ResizableBag in your implementation as defined in the UML diagram below. You only need to finish the Billiard.java class that contains main. Please note the **sample run** below.

### UML Diagram:



**SAMPLE RUN:**What is the first numbered ball to start with? (must be greater than 0)  
**5**

The first ball is: "5"

\*\*\* Removing balls from the table \*\*\*

--> Removed "5"

After adding 5 balls, we have 5 balls on the table:

[1, 1, 4, 3, 4]

--> Removed "4"

After adding 4 balls, we have 8 balls on the table:

[1, 1, 4, 3, 3, 1, 2, 3]

--> Removed "3"

After adding 3 balls, we have 10 balls on the table:

[1, 1, 4, 3, 3, 1, 2, 1, 1, 1]

--> Removed "1"

Removed ball has number "1", no new balls will be added - 9 balls remaining.

[1, 1, 4, 3, 3, 1, 2, 1, 1]

--> Removed "1"

Removed ball has number "1", no new balls will be added - 8 balls remaining.

[1, 1, 4, 3, 3, 1, 2, 1]

--> Removed "1"

Removed ball has number "1", no new balls will be added - 7 balls remaining.

[1, 1, 4, 3, 3, 1, 2]

--> Removed "2"

After adding 2 balls, we have 8 balls on the table:

[1, 1, 4, 3, 3, 1, 1, 1]

--> Removed "1"

Removed ball has number "1", no new balls will be added - 7 balls remaining.

[1, 1, 4, 3, 3, 1, 1]

--> Removed "1"

Removed ball has number "1", no new balls will be added - 6 balls remaining.

[1, 1, 4, 3, 3, 1]

--> Removed "1"

Removed ball has number "1", no new balls will be added - 5 balls remaining.

[1, 1, 4, 3, 3]

--> Removed "3"

After adding 3 balls, we have 7 balls on the table:

[1, 1, 4, 3, 1, 1, 2]

--> Removed "2"

After adding 2 balls, we have 8 balls on the table:

[1, 1, 4, 3, 1, 1, 1, 1]

--> Removed "1"

Removed ball has number "1", no new balls will be added - 7 balls remaining.

[1, 1, 4, 3, 1, 1, 1]

--> Removed "1"

Removed ball has number "1", no new balls will be added - 6 balls remaining.

[1, 1, 4, 3, 1, 1]

--> Removed "1"

Removed ball has number "1", no new balls will be added - 5 balls remaining.

[1, 1, 4, 3, 1]

--> Removed "1"

Removed ball has number "1", no new balls will be added - 4 balls remaining.

[1, 1, 4, 3]

--> Removed "3"

After adding 3 balls, we have 6 balls on the table:

[1, 1, 4, 1, 2, 1]

--> Removed "1"

Removed ball has number "1", no new balls will be added - 5 balls remaining.

[1, 1, 4, 1, 2]

--> Removed "2"

After adding 2 balls, we have 6 balls on the table:

[1, 1, 4, 1, 1, 1]

--> Removed "1"

Removed ball has number "1", no new balls will be added - 5 balls remaining.

[1, 1, 4, 1, 1]

--> Removed "1"

Removed ball has number "1", no new balls will be added - 4 balls remaining.

[1, 1, 4, 1]

--> Removed "1"

Removed ball has number "1", no new balls will be added - 3 balls remaining.

[1, 1, 4]

--> Removed "4"

After adding 4 balls, we have 6 balls on the table:

[1, 1, 3, 1, 2, 2]

--> Removed "2"

After adding 2 balls, we have 7 balls on the table:

[1, 1, 3, 1, 2, 1, 1]

--> Removed "1"

Removed ball has number "1", no new balls will be added - 6 balls remaining.

[1, 1, 3, 1, 2, 1]

--> Removed "1"

Removed ball has number "1", no new balls will be added - 5 balls remaining.

[1, 1, 3, 1, 2]

--> Removed "2"

After adding 2 balls, we have 6 balls on the table:

[1, 1, 3, 1, 1, 1]

--> Removed "1"

Removed ball has number "1", no new balls will be added - 5 balls remaining.

[1, 1, 3, 1, 1]

--> Removed "1"

Removed ball has number "1", no new balls will be added - 4 balls remaining.

[1, 1, 3, 1]

--> Removed "1"

Removed ball has number "1", no new balls will be added - 3 balls remaining.

[1, 1, 3]

--> Removed "3"

After adding 3 balls, we have 5 balls on the table:

[1, 1, 1, 1, 1]

--> Removed "1"

Removed ball has number "1", no new balls will be added - 4 balls remaining.

[1, 1, 1, 1]

--> Removed "1"

Removed ball has number "1", no new balls will be added - 3 balls remaining.

[1, 1, 1]

--> Removed "1"

Removed ball has number "1", no new balls will be added - 2 balls remaining.

[1, 1]

--> Removed "1"

Removed ball has number "1", no new balls will be added - 1 balls remaining.

[1]

--> Removed "1"

Removed ball has number "1", no new balls will be added - 0 balls remaining.

[]

The table is empty!!!

The time required was 5 milliseconds

Process finished with exit code 0

**Project2**

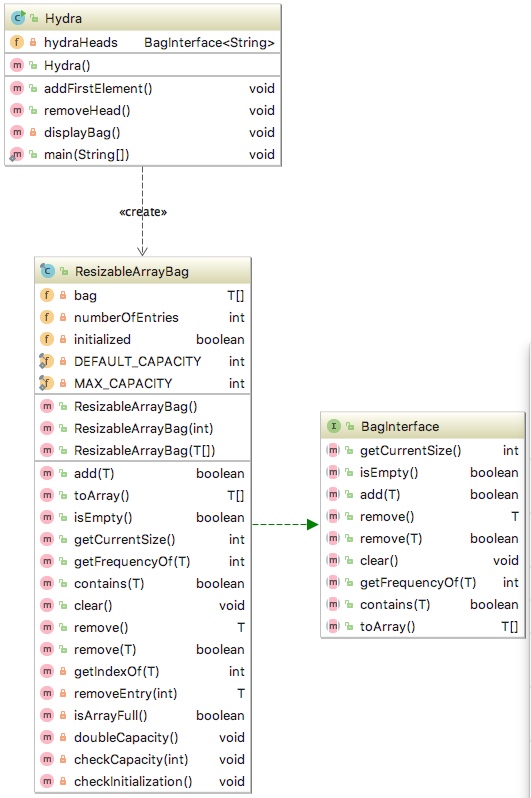
In mythology, *Hydra of Lerna* was a monster with many heads. Every time the hero chopped off a head, two smaller heads would grow in its place. Fortunately for the hero, if the head was small enough, he could chop it off without two more growing in its place. To kill the Hydra, all our hero needed to do was to chop off all the heads.



Write a program that simulates the Hydra. We will be using strings to represent heads. A bag of strings, then, represents the Hydra. Every time you remove a string from the bag, delete the first letter of the string and put two copies of the remaining string back into a bag. For example, if you remove HYDRA, you add two copies of YDRA to the bag. If you remove a one-letter string, you add nothing to the bag (see Sample Run below).

To begin, read one word from the keyboard and place it into an empty bag. The Hydra dies when the bag becomes empty. Follow the design and implementation of project1 (see the UML diagram below).

### UML Diagram:



**SAMPLE RUN:**

What is the initial string?  
**HYDRA**

The initial string "HYDRA" has length of 5

\*\*\* Removing heads from the Hydra \*\*\*

--> Removed "HYDRA"

After adding two, the Hydra has 2 heads:

[YDRA, YDRA]

--> Removed "YDRA"

After adding two, the Hydra has 3 heads:

[YDRA, DRA, DRA]

--> Removed "DRA"

After adding two, the Hydra has 4 heads:

[YDRA, DRA, RA, RA]

--> Removed "RA"

After adding two, the Hydra has 5 heads:

[YDRA, DRA, RA, A, A]

--> Removed "A"

The removed head is of length 1, no new heads will be added - 4 heads remaining

[YDRA, DRA, RA, A]

--> Removed "A"

The removed head is of length 1, no new heads will be added - 3 heads remaining

[YDRA, DRA, RA]

--> Removed "RA"

After adding two, the Hydra has 4 heads:

[YDRA, DRA, A, A]

--> Removed "A"

The removed head is of length 1, no new heads will be added - 3 heads remaining

[YDRA, DRA, A]

--> Removed "A"

The removed head is of length 1, no new heads will be added - 2 heads remaining

[YDRA, DRA]

--> Removed "DRA"

After adding two, the Hydra has 3 heads:

[YDRA, RA, RA]

--> Removed "RA"

After adding two, the Hydra has 4 heads:

[YDRA, RA, A, A]

--> Removed "A"

The removed head is of length 1, no new heads will be added - 3 heads remaining

[YDRA, RA, A]

--> Removed "A"

The removed head is of length 1, no new heads will be added - 2 heads remaining

[YDRA, RA]

--> Removed "RA"

After adding two, the Hydra has 3 heads:

[YDRA, A, A]

--> Removed "A"

The removed head is of length 1, no new heads will be added - 2 heads remaining

[YDRA, A]

--> Removed "A"

The removed head is of length 1, no new heads will be added - 1 heads remaining

[YDRA]

--> Removed "YDRA"

After adding two, the Hydra has 2 heads:

[DRA, DRA]

--> Removed "DRA"

After adding two, the Hydra has 3 heads:

[DRA, RA, RA]

--> Removed "RA"

After adding two, the Hydra has 4 heads:

[DRA, RA, A, A]

--> Removed "A"

The removed head is of length 1, no new heads will be added - 3 heads remaining

[DRA, RA, A]

--> Removed "A"

The removed head is of length 1, no new heads will be added - 2 heads remaining

[DRA, RA]

--> Removed "RA"

After adding two, the Hydra has 3 heads:

[DRA, A, A]

--> Removed "A"

The removed head is of length 1, no new heads will be added - 2 heads remaining

[DRA, A]

--> Removed "A"

The removed head is of length 1, no new heads will be added - 1 heads remaining

[DRA]

--> Removed "DRA"

After adding two, the Hydra has 2 heads:

[RA, RA]

--> Removed "RA"

After adding two, the Hydra has 3 heads:

[RA, A, A]

--> Removed "A"

The removed head is of length 1, no new heads will be added - 2 heads remaining

[RA, A]

--> Removed "A"

The removed head is of length 1, no new heads will be added - 1 heads remaining

[RA]

--> Removed "RA"

After adding two, the Hydra has 2 heads:

[A, A]

--> Removed "A"

The removed head is of length 1, no new heads will be added - 1 heads remaining

[A]

--> Removed "A"

The removed head is of length 1, no new heads will be added - 0 heads remaining

[]

The Hydra is no more!!!

The time required was 4 milliseconds

Process finished with exit code 0

**Project3**

Consider an abstract class Money with two subclasses:

* class Coin that represents a coin
* class Bill that represents a paper bill.

A **paper bill** can have one of the 7 denomination values: 0, 1, 2, 3, 4, 5, 6 representing respectively: 1, 2, 5, 10, 20, 50, and 100 dollars with names: WASHINGTON, JEFFERSON, LINCOLN, HAMILTON, JACKSON, GRANT, and FRANKLIN respectively.

A **coin** can have one of the 5 denomination values: 0, 1, 2, 3, 4 representing respectively: 1, 5, 10, 25, and 50 cents with names: PENNY, NICKEL, DIME, QUARTER, and HALF\_DOLLAR respectively.

Denominations and denomination names are stored in respective arrays.

Each coin and bill can either be placed HEADS or TAILS up.

Money constructor generates the denomination randomly based on the number of denominations passed to it and sets the value of heads to false.

The method getValue returns the value of a coin (for example 0.25 for a quarter) or the value of paper bill respectively.

The method toss simulates a money toss in which the coin/paper bill lands either heads up or tails up.

The method isHeads returns true if a coin is heads up.

The class has only one constructor, the secondary constructor, which sets the value of heads to false; the denomination is randomly generated based on the numberOfDenominations passed to it.

Now, you need to have a piggy bank with predefined capacity to hold your monies. The piggy bank holds the monies but gives them no other organization. And it can certainly contain duplicates. The piggy bank is an ADT that has only the following operations:

* check the capacity of the piggy bank,
* you can add money to the piggy bank one at a time,
  + adding fails if there is no more room in the piggy bank (PiggyBankFullException will be thrown that must be handled by the client)
* remove one (randomly selected),
  + remove fails if the piggy bank is empty (PiggyBankEmptyException will be thrown that must be handled by the client)
* see whether the piggy bank is empty,
* see whether the piggy bank is full,
* see how many monies can fit in the piggy bank,
* see how many monies are in the piggy bank,
* see the content of the piggy bank,
* shake the piggy bank to rearrange the monies,
* empty the piggy bank counting how many monies landed HEADS

This functionality should be implemented in a class PiggyBank where the piggy bank is represented as an ResizableArrayBag of Money objects (this.bucket instance variable). The PiggyBank constructor creates this.bucket object and fills it with the given number of monies, randomly choosing a Coin or a Bill object to create and add to this.bucket.

PLEASE NOTE:

* that shake method must utilize toArray method to get the access to the objects:

Object[] toShake = this.bucket.toArray();

For each element in the toShake array select randomly another element to swap it with. Once the objects are shuffled in the toShake array they need to be added to this.bucket (which needs to be cleared first)

* PiggyBank toString method utilizes **polymorphism**:
  + Calls toArray to get the elements (see above)
  + Uses a for loop over the resulted array to compute the total - must use casting: total += ((Money) content[i]).getValue();
  + Calls Arrays.toString to get the content of the objects

emptyAndCountHeads method while removing money from the piggy bank counts how many coins/bills landed HEADS, displays the count and the dollar value, and returns the heads count.

### Client called CountHeadsGame is fully implemented. It manages a game where the user and the computer take turns to create the piggy bank with the same capacity and the same initial number of monies (since the monies are randomly generated the denominations will be different for each player). Each player empties the piggy bank. The player with the larger number of monies landing HEADS wins. The tie is also possible.

Round constructor creates PiggyBank object and displays the content of the piggy bank. The Round object is created in the main.

addTwoMonies method creates one Coin object and one Bill object and attempts to add them to the piggy bank. Since the piggy bank may be full, the addTwoMonies method must handle PiggyBankFullException

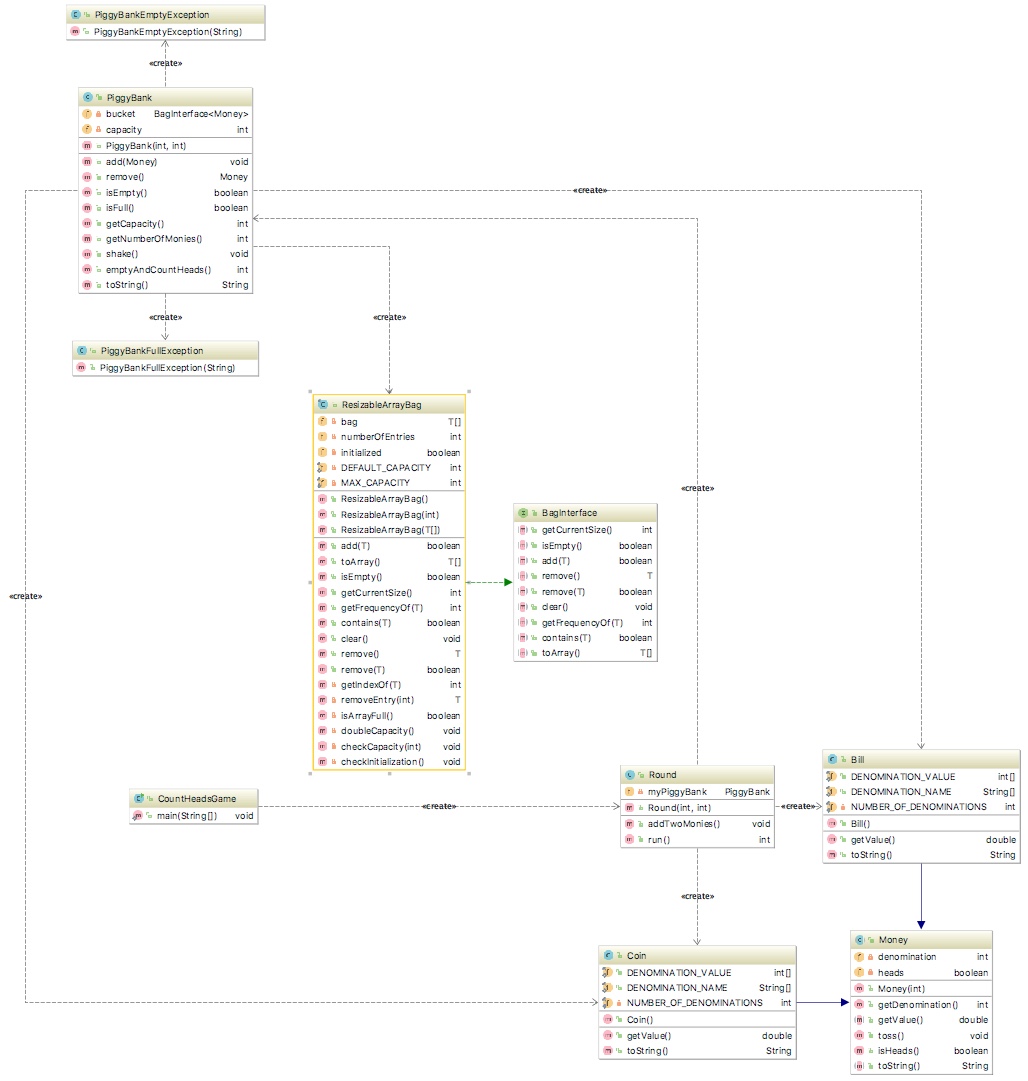
run method calls addTwoMonies method; shakes the piggy bank; prints its content; asks the piggy bank to emptyAndCountHeads and returns the number of heads.

### See the “Sample Runs” below.

### Your Task:

1. Load the provided classes into IDEA
2. Create all the remaining classes as defined in the UML diagram provided below:
   1. Define all the methods as skeletons first (see Round class)
   2. Declare all instance variables private
   3. Pay attention which methods should be defined as abstract
   4. Make sure to implement inheritance relationship
   5. Pay attention which variables should be defined as static
   6. Pay attention which variables should be defined as final
   7. Make sure that all constants have appropriate values assigned at the declaration level
   8. Do not start implementation until all classes compile as skeletons
3. Use javadoc-style comments to describe the purpose of each method, its parameters and return values.
4. Implement the application iteratively using the bottom-up approach by implementing the classes in the following order:
   1. Money
   2. Bill
   3. Coin
   4. PiggyBank
   5. Round
5. Make sure that the output is properly formatted as shown in the “Sample Runs” below.

### UML Diagram:



### Sample Runs:

#### Run#1:

#### How many coins/bills can fit in your piggy bank?

#### 10

#### How many coins/bills should the computer put in your piggy bank?

#### 60

#### How many rounds should we play?

#### 4

#### How many coins/bills can fit in your piggy bank?

#### 10

#### How many coins/bills should the computer put in your piggy bank?

#### 6

#### How many rounds should we play?

#### 4

#### \*\*\*\*\*\*\* Round #1 --> USER'S TURN \*\*\*\*\*\*\*

#### >> Adding 6 monies to your piggy bank <<

#### Added $0.50 to the piggy bank

#### Added $0.05 to the piggy bank

#### Added $0.10 to the piggy bank

#### Added $20.00 to the piggy bank

#### Added $10.00 to the piggy bank

#### Added $0.01 to the piggy bank

#### >> The content of your piggy bank <<

#### The piggy bank can hold 10 coins/bills;

#### There are 6 coins/bills in the piggy bank:[HALF\_DOLLAR landed TAILS, NICKEL landed TAILS, DIME landed TAILS, JACKSON landed TAILS, HAMILTON landed TAILS, PENNY landed TAILS]

#### The total of $30.66

#### --> Adding additional monies:

#### Added $0.25 to the piggy bank

#### Added $50.00 to the piggy bank

#### >> Shaking your piggy bank <<

#### The piggy bank can hold 10 coins/bills;

#### There are 8 coins/bills in the piggy bank:[PENNY landed TAILS, HAMILTON landed TAILS, GRANT landed TAILS, DIME landed TAILS, NICKEL landed TAILS, QUARTER landed TAILS, JACKSON landed TAILS, HALF\_DOLLAR landed TAILS]

#### The total of $80.91

#### >> Emptying your piggy bank <<

#### HALF\_DOLLAR landed HEADS

#### JACKSON landed HEADS

#### QUARTER landed TAILS

#### NICKEL landed TAILS

#### DIME landed HEADS

#### GRANT landed HEADS

#### HAMILTON landed HEADS

#### PENNY landed TAILS

#### 5 out of 8 coins/bills landed "HEADS"

#### The total value of "HEADS" is: $80.60

#### \*\*\*\*\*\*\* Round #2 --> COMPUTER'S TURN \*\*\*\*\*\*\*

#### >> Adding 6 monies to your piggy bank <<

#### Added $0.05 to the piggy bank

#### Added $0.05 to the piggy bank

#### Added $0.05 to the piggy bank

#### Added $0.10 to the piggy bank

#### Added $2.00 to the piggy bank

#### Added $20.00 to the piggy bank

#### >> The content of your piggy bank <<

#### The piggy bank can hold 10 coins/bills;

#### There are 6 coins/bills in the piggy bank:[NICKEL landed TAILS, NICKEL landed TAILS, NICKEL landed TAILS, DIME landed TAILS, JEFFERSON landed TAILS, JACKSON landed TAILS]

#### The total of $22.25

#### --> Adding additional monies:

#### Added $0.01 to the piggy bank

#### Added $5.00 to the piggy bank

#### >> Shaking your piggy bank <<

#### The piggy bank can hold 10 coins/bills;

#### There are 8 coins/bills in the piggy bank:[PENNY landed TAILS, DIME landed TAILS, NICKEL landed TAILS, LINCOLN landed TAILS, NICKEL landed TAILS, JEFFERSON landed TAILS, NICKEL landed TAILS, JACKSON landed TAILS]

#### The total of $27.26

#### >> Emptying your piggy bank <<

#### JACKSON landed HEADS

#### NICKEL landed HEADS

#### JEFFERSON landed TAILS

#### NICKEL landed TAILS

#### LINCOLN landed TAILS

#### NICKEL landed HEADS

#### DIME landed HEADS

#### PENNY landed TAILS

#### 4 out of 8 coins/bills landed "HEADS"

#### The total value of "HEADS" is: $20.20

#### \*\*\*\*\*\*\* Round #3 --> USER'S TURN \*\*\*\*\*\*\*

#### >> Adding 6 monies to your piggy bank <<

#### Added $0.25 to the piggy bank

#### Added $0.05 to the piggy bank

#### Added $2.00 to the piggy bank

#### Added $50.00 to the piggy bank

#### Added $10.00 to the piggy bank

#### Added $0.25 to the piggy bank

#### >> The content of your piggy bank <<

#### The piggy bank can hold 10 coins/bills;

#### There are 6 coins/bills in the piggy bank:[QUARTER landed TAILS, NICKEL landed TAILS, JEFFERSON landed TAILS, GRANT landed TAILS, HAMILTON landed TAILS, QUARTER landed TAILS]

#### The total of $62.55

#### --> Adding additional monies:

#### Added $0.10 to the piggy bank

#### Added $50.00 to the piggy bank

#### >> Shaking your piggy bank <<

#### The piggy bank can hold 10 coins/bills;

#### There are 8 coins/bills in the piggy bank:[GRANT landed TAILS, QUARTER landed TAILS, GRANT landed TAILS, DIME landed TAILS, JEFFERSON landed TAILS, NICKEL landed TAILS, QUARTER landed TAILS, HAMILTON landed TAILS]

#### The total of $112.65

#### >> Emptying your piggy bank <<

#### HAMILTON landed TAILS

#### QUARTER landed HEADS

#### NICKEL landed HEADS

#### JEFFERSON landed HEADS

#### DIME landed TAILS

#### GRANT landed HEADS

#### QUARTER landed TAILS

#### GRANT landed HEADS

#### 5 out of 8 coins/bills landed "HEADS"

#### The total value of "HEADS" is: $102.30

#### \*\*\*\*\*\*\* Round #4 --> COMPUTER'S TURN \*\*\*\*\*\*\*

#### >> Adding 6 monies to your piggy bank <<

#### Added $0.01 to the piggy bank

#### Added $100.00 to the piggy bank

#### Added $10.00 to the piggy bank

#### Added $20.00 to the piggy bank

#### Added $0.01 to the piggy bank

#### Added $100.00 to the piggy bank

#### >> The content of your piggy bank <<

#### The piggy bank can hold 10 coins/bills;

#### There are 6 coins/bills in the piggy bank:[PENNY landed TAILS, FRANKLIN landed TAILS, HAMILTON landed TAILS, JACKSON landed TAILS, PENNY landed TAILS, FRANKLIN landed TAILS]

#### The total of $230.02

#### --> Adding additional monies:

#### Added $0.50 to the piggy bank

#### Added $10.00 to the piggy bank

#### >> Shaking your piggy bank <<

#### The piggy bank can hold 10 coins/bills;

#### There are 8 coins/bills in the piggy bank:[PENNY landed TAILS, FRANKLIN landed TAILS, FRANKLIN landed TAILS, JACKSON landed TAILS, PENNY landed TAILS, HALF\_DOLLAR landed TAILS, HAMILTON landed TAILS, HAMILTON landed TAILS]

#### The total of $240.52

#### >> Emptying your piggy bank <<

#### HAMILTON landed HEADS

#### HAMILTON landed TAILS

#### HALF\_DOLLAR landed HEADS

#### PENNY landed HEADS

#### JACKSON landed HEADS

#### FRANKLIN landed HEADS

#### FRANKLIN landed TAILS

#### PENNY landed HEADS

#### 6 out of 8 coins/bills landed "HEADS"

#### The total value of "HEADS" is: $130.52

#### \*\*\*GAME OVER\*\*\*

#### computerHeadsCount = 10

#### userHeadsCount = 10

#### It's a tie!!!

#### Process finished with exit code 0

#### Run#2:

How many coins/bills can fit in your piggy bank?

**10**

How many coins/bills should the computer put in your piggy bank?

**9**

How many rounds should we play?

**2**

\*\*\*\*\*\*\* Round #1 --> USER'S TURN \*\*\*\*\*\*\*

>> Adding 9 monies to your piggy bank <<

Added $20.00 to the piggy bank

Added $1.00 to the piggy bank

Added $0.50 to the piggy bank

Added $1.00 to the piggy bank

Added $0.10 to the piggy bank

Added $0.50 to the piggy bank

Added $50.00 to the piggy bank

Added $2.00 to the piggy bank

Added $0.25 to the piggy bank

>> The content of your piggy bank <<

The piggy bank can hold 10 coins/bills;

There are 9 coins/bills in the piggy bank:[JACKSON landed TAILS, WASHINGTON landed TAILS, HALF\_DOLLAR landed TAILS, WASHINGTON landed TAILS, DIME landed TAILS, HALF\_DOLLAR landed TAILS, GRANT landed TAILS, JEFFERSON landed TAILS, QUARTER landed TAILS]

The total of $75.35

--> Adding additional monies:

Added $0.10 to the piggy bank

No more room in the piggy bank! - additional monies will not be added to your piggy bank.

>> Shaking your piggy bank <<

The piggy bank can hold 10 coins/bills;

There are 10 coins/bills in the piggy bank:[DIME landed TAILS, QUARTER landed TAILS, JEFFERSON landed TAILS, GRANT landed TAILS, HALF\_DOLLAR landed TAILS, JACKSON landed TAILS, DIME landed TAILS, HALF\_DOLLAR landed TAILS, WASHINGTON landed TAILS, WASHINGTON landed TAILS]

The total of $75.45

>> Emptying your piggy bank <<

WASHINGTON landed HEADS

WASHINGTON landed HEADS

HALF\_DOLLAR landed HEADS

DIME landed TAILS

JACKSON landed TAILS

HALF\_DOLLAR landed TAILS

GRANT landed HEADS

JEFFERSON landed TAILS

QUARTER landed TAILS

DIME landed TAILS

4 out of 10 coins/bills landed "HEADS"

The total value of "HEADS" is: $52.50

\*\*\*\*\*\*\* Round #2 --> COMPUTER'S TURN \*\*\*\*\*\*\*

>> Adding 9 monies to your piggy bank <<

Added $0.50 to the piggy bank

Added $0.50 to the piggy bank

Added $10.00 to the piggy bank

Added $0.25 to the piggy bank

Added $0.05 to the piggy bank

Added $0.10 to the piggy bank

Added $0.05 to the piggy bank

Added $100.00 to the piggy bank

Added $0.10 to the piggy bank

>> The content of your piggy bank <<

The piggy bank can hold 10 coins/bills;

There are 9 coins/bills in the piggy bank:[HALF\_DOLLAR landed TAILS, HALF\_DOLLAR landed TAILS, HAMILTON landed TAILS, QUARTER landed TAILS, NICKEL landed TAILS, DIME landed TAILS, NICKEL landed TAILS, FRANKLIN landed TAILS, DIME landed TAILS]

The total of $111.55

--> Adding additional monies:

Added $0.25 to the piggy bank

No more room in the piggy bank! - additional monies will not be added to your piggy bank.

>> Shaking your piggy bank <<

The piggy bank can hold 10 coins/bills;

There are 10 coins/bills in the piggy bank:[DIME landed TAILS, NICKEL landed TAILS, NICKEL landed TAILS, FRANKLIN landed TAILS, HALF\_DOLLAR landed TAILS, QUARTER landed TAILS, HALF\_DOLLAR landed TAILS, DIME landed TAILS, QUARTER landed TAILS, HAMILTON landed TAILS]

The total of $111.80

>> Emptying your piggy bank <<

HAMILTON landed HEADS

QUARTER landed TAILS

DIME landed HEADS

HALF\_DOLLAR landed HEADS

QUARTER landed TAILS

HALF\_DOLLAR landed HEADS

FRANKLIN landed HEADS

NICKEL landed TAILS

NICKEL landed HEADS

DIME landed TAILS

6 out of 10 coins/bills landed "HEADS"

The total value of "HEADS" is: $111.15

\*\*\*GAME OVER\*\*\*

computerHeadsCount = 6

userHeadsCount = 4

Computer wins!!!

Process finished with exit code 0

#### Run#3:

How many coins/bills can fit in your piggy bank?

**10**

How many coins/bills should the computer put in your piggy bank?

**8**

How many rounds should we play?

\*\*\*\*\*\*\* Round #1 --> COMPUTER'S TURN \*\*\*\*\*\*\*

>> Adding 8 monies to your piggy bank <<

Added $2.00 to the piggy bank

Added $0.50 to the piggy bank

Added $0.01 to the piggy bank

Added $0.50 to the piggy bank

Added $50.00 to the piggy bank

Added $0.50 to the piggy bank

Added $100.00 to the piggy bank

Added $50.00 to the piggy bank

>> The content of your piggy bank <<

The piggy bank can hold 10 coins/bills;

There are 8 coins/bills in the piggy bank:[JEFFERSON landed TAILS, HALF\_DOLLAR landed TAILS, PENNY landed TAILS, HALF\_DOLLAR landed TAILS, GRANT landed TAILS, HALF\_DOLLAR landed TAILS, FRANKLIN landed TAILS, GRANT landed TAILS]

The total of $203.51

--> Adding additional monies:

Added $0.10 to the piggy bank

Added $5.00 to the piggy bank

>> Shaking your piggy bank <<

The piggy bank can hold 10 coins/bills;

There are 10 coins/bills in the piggy bank:[DIME landed TAILS, HALF\_DOLLAR landed TAILS, GRANT landed TAILS, PENNY landed TAILS, HALF\_DOLLAR landed TAILS, FRANKLIN landed TAILS, GRANT landed TAILS, JEFFERSON landed TAILS, LINCOLN landed TAILS, HALF\_DOLLAR landed TAILS]

The total of $208.61

>> Emptying your piggy bank <<

HALF\_DOLLAR landed TAILS

LINCOLN landed TAILS

JEFFERSON landed HEADS

GRANT landed HEADS

FRANKLIN landed HEADS

HALF\_DOLLAR landed TAILS

PENNY landed HEADS

GRANT landed TAILS

HALF\_DOLLAR landed TAILS

DIME landed HEADS

5 out of 10 coins/bills landed "HEADS"

The total value of "HEADS" is: $152.11

\*\*\*\*\*\*\* Round #2 --> USER'S TURN \*\*\*\*\*\*\*

>> Adding 8 monies to your piggy bank <<

Added $5.00 to the piggy bank

Added $0.01 to the piggy bank

Added $2.00 to the piggy bank

Added $2.00 to the piggy bank

Added $0.50 to the piggy bank

Added $0.25 to the piggy bank

Added $20.00 to the piggy bank

Added $5.00 to the piggy bank

>> The content of your piggy bank <<

The piggy bank can hold 10 coins/bills;

There are 8 coins/bills in the piggy bank:[LINCOLN landed TAILS, PENNY landed TAILS, JEFFERSON landed TAILS, JEFFERSON landed TAILS, HALF\_DOLLAR landed TAILS, QUARTER landed TAILS, JACKSON landed TAILS, LINCOLN landed TAILS]

The total of $34.76

--> Adding additional monies:

Added $0.10 to the piggy bank

Added $10.00 to the piggy bank

>> Shaking your piggy bank <<

The piggy bank can hold 10 coins/bills;

There are 10 coins/bills in the piggy bank:[JEFFERSON landed TAILS, QUARTER landed TAILS, JACKSON landed TAILS, LINCOLN landed TAILS, PENNY landed TAILS, DIME landed TAILS, HALF\_DOLLAR landed TAILS, HAMILTON landed TAILS, LINCOLN landed TAILS, JEFFERSON landed TAILS]

The total of $44.86

>> Emptying your piggy bank <<

JEFFERSON landed HEADS

LINCOLN landed TAILS

HAMILTON landed HEADS

HALF\_DOLLAR landed HEADS

DIME landed HEADS

PENNY landed HEADS

LINCOLN landed TAILS

JACKSON landed HEADS

QUARTER landed TAILS

JEFFERSON landed TAILS

6 out of 10 coins/bills landed "HEADS"

The total value of "HEADS" is: $32.61

\*\*\*GAME OVER\*\*\*

computerHeadsCount = 5

userHeadsCount = 6

User wins!!!

Process finished with exit code 0