

CSE1341 - Lab 6 Assignment

Overview

In Lab 5 you created a SlotMachine using structured programming techniques. In this lab, you will rewrite your application using object-oriented programming.

Pre-Lab (5 Points)

Create the GameLauncher class with a main method that includes the code provided in the instructions. Bring this to your lab session for credit.

Lab (95 Points)

Create the SlotMachine system using object oriented programming, following the design provided on the following pages. Your output should match the format shown on the last page, although your actual output will vary based on the the outcome of the game.



NOTES:

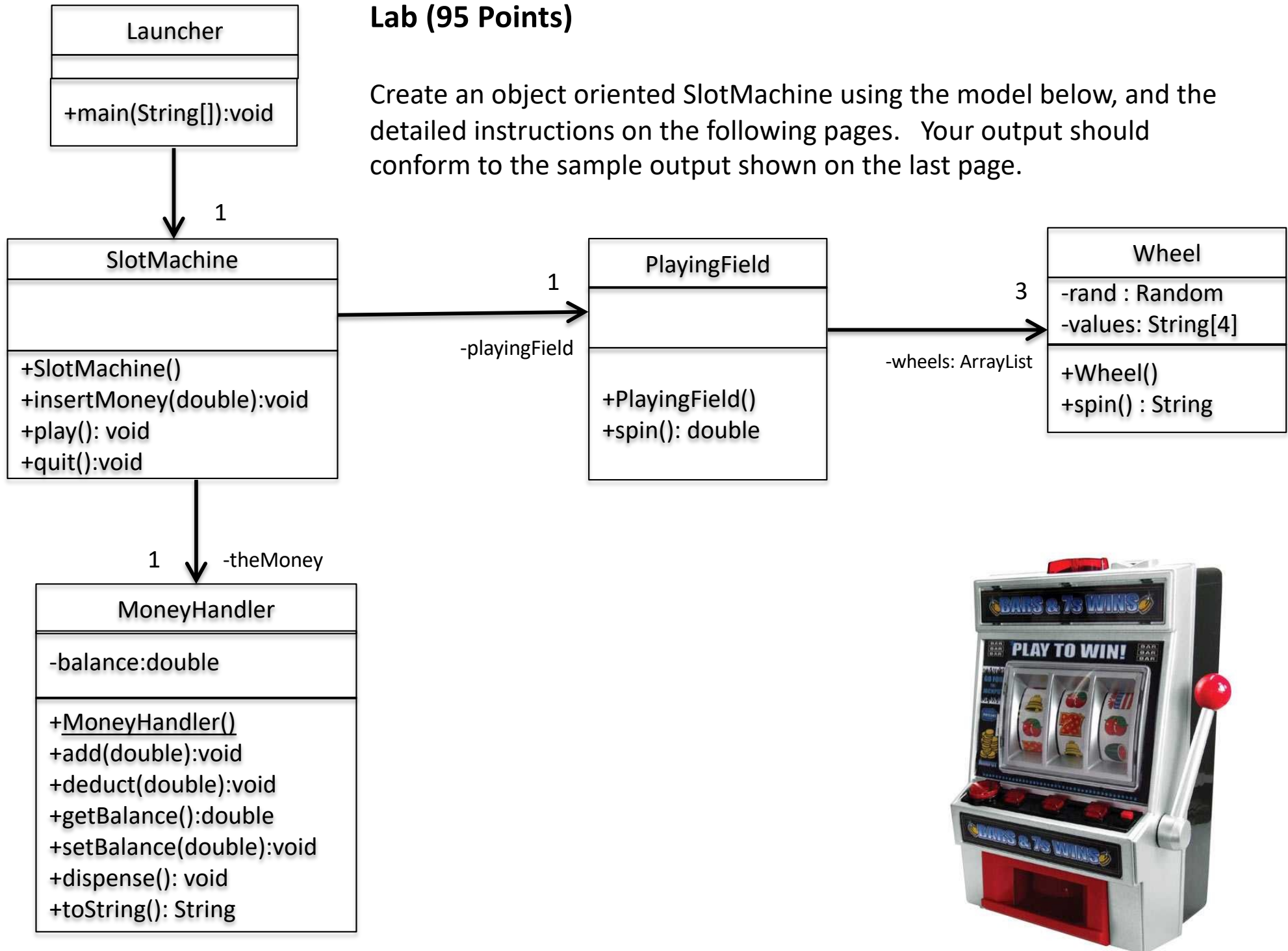
Each program should include comments that explain what each block of code is doing. Additionally, the programs should compile without errors, and run with the results described in the exercise. The following deductions will be made from each exercise if any of the following is incorrect or missing:

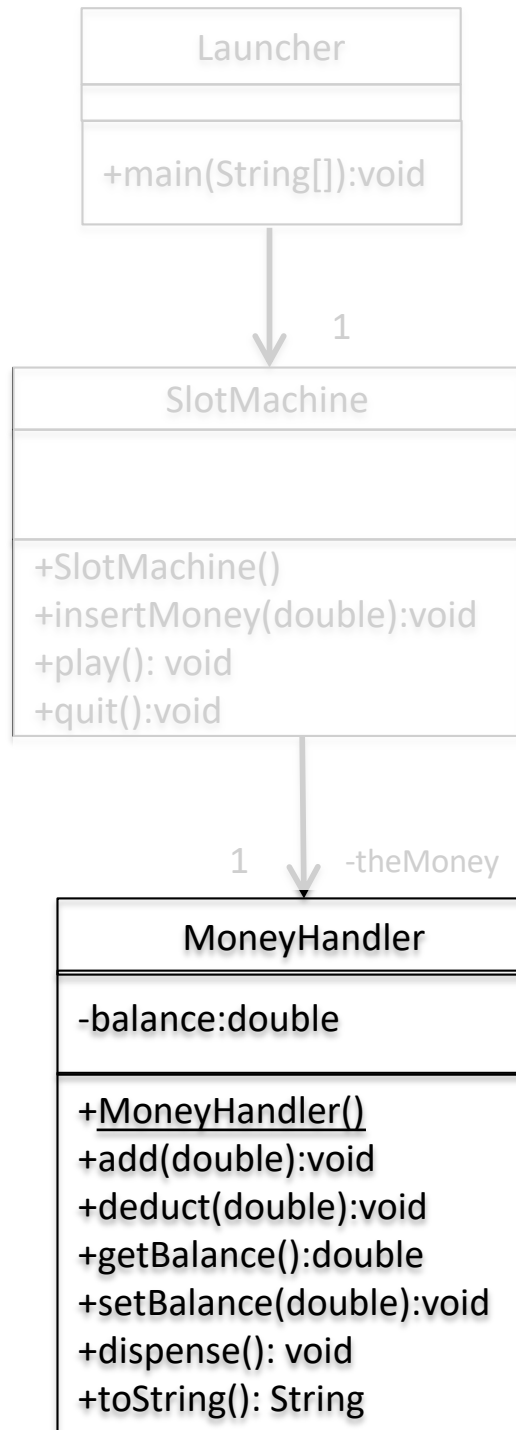
- Proper formatting [5 points]
- Proper names for classes and variables [5 points]
- Comments [5 points per class]
- Program doesn't compile [10 points]
- Source code (java file) missing [10 points]
- Executable (class file) missing [10 points]
- Missing array where an array was required [5 points each]
- Missing loop where a loop was required [5 points each]
- Missing class from the design provided [10 points each]
- Missing method from the design provided [5 points each]

This Lab is due Saturday November 5, 2022 at 6:00am.

Lab (95 Points)

Create an object oriented SlotMachine using the model below, and the detailed instructions on the following pages. Your output should conform to the sample output shown on the last page.





MoneyHandler methods: [25 points]

Constructor

- Initialize balance to 0

add

- Adds the value passed in to the value of balance

deduct

- Deducts the value passed in from the balance

getBalance

- Getter for the the balance attribute

setBalance

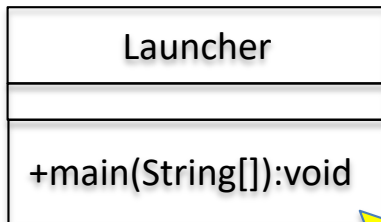
- Setter for the the balance attribute

dispense

- Prints a "dispensed" message with the balance
- Sets the value of balance to zero

toString

- Returns a String with text containing the current balance



Launcher: [created for pre-lab]

Put the following code in the *main* method:

```
SlotMachine slot = new SlotMachine();
Scanner input = new Scanner(System.in);
System.out.print("How much money would you like to insert? ");
double money = input.nextDouble();
slot.insertMoney(money);

while(true)
{
    System.out.print("Spin/Quit (S/Q) : ");
    String s = input.next();
    if(s.equalsIgnoreCase("Q"))
        break;
    else
        slot.play();
}
slot.quit();
```

Launcher

+main(String[] args)

Wheel: [20 points]

Each wheel contains an array with four Strings (Micky, Minnie, Donald, Daisy). This array and its contents are created in the Wheel constructor.

The *spin* method generates a random number. Use this as an index number in the array to select and return one of the four String values.

SlotMachine

+SlotMachine()
+insertMoney()
+play(): void
+quit(): void

1

-theMoney

MoneyHandler

-balance: double

+MoneyHandler()
+add(double): void
+deduct(double): void
+getBalance(): double
+setBalance(double): void
+dispense(): void
+toString(): String

Wheel

-values: String[4]
-rand : Random

+Wheel()
+spin(): String

3

-wheels: ArrayList



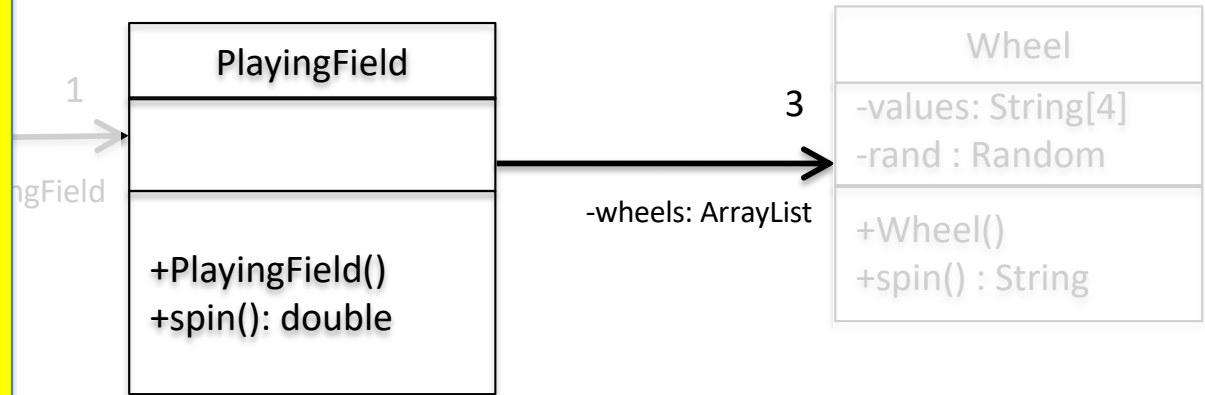
PlayingField: [30 points]

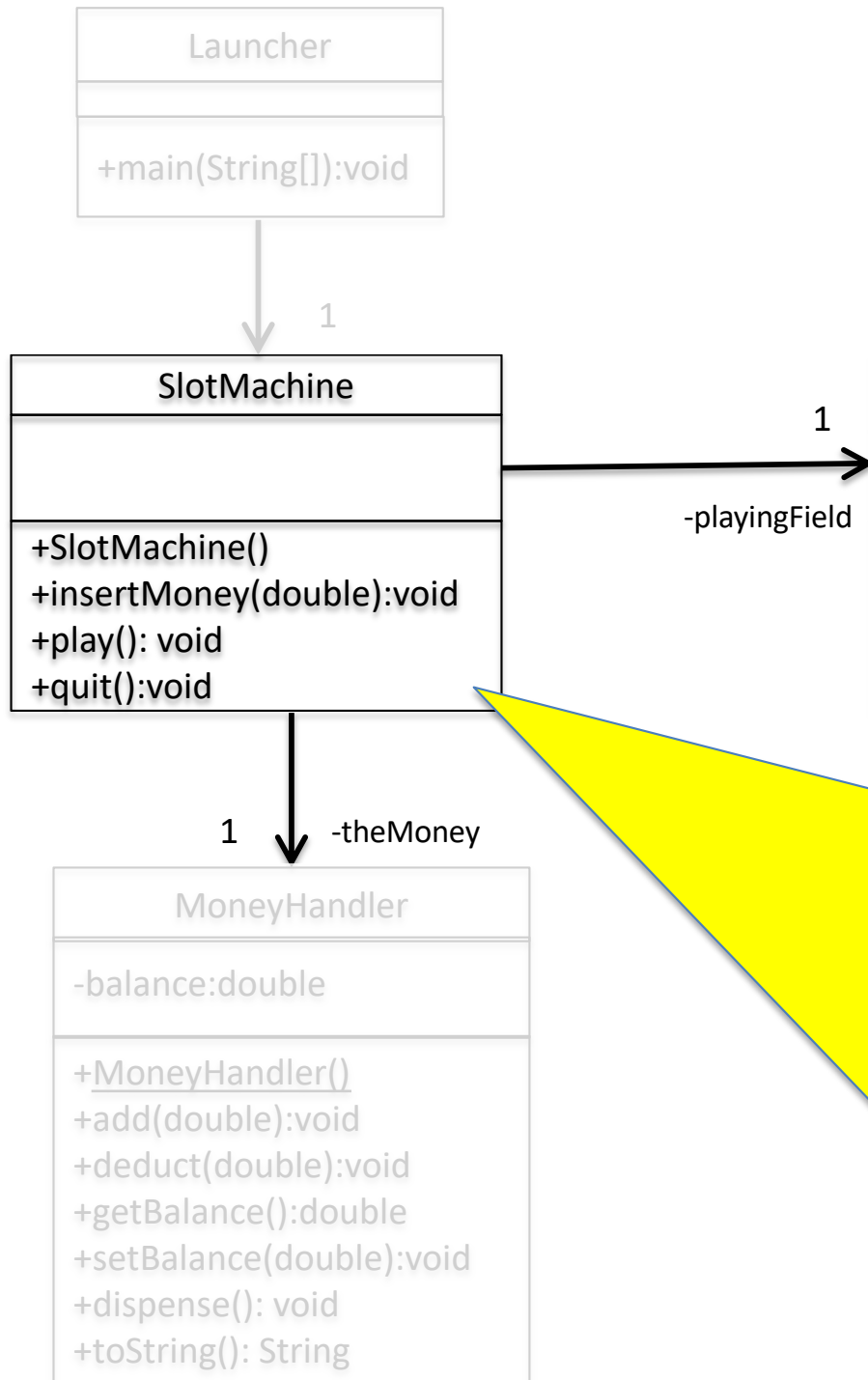
Constructor

Create three Wheel objects and keeps them in an ArrayList named *wheels*.

spin

- send the *spin* method to each of the Wheel objects in the *wheels* ArrayList. Keep the returned values in String variables. Print a message with the three Strings to show what was spun.
- These Strings are compared to determine the prize amount to return:
- If all three items match, return 1.00 If two items match, return 0.50 Otherwise, return 0.0





SlotMachine: [20 points]

Constructor

- Creates playingField and MoneyHandler objects and retains them in the attributes shown in the model.

insertMoney

- Accepts a double and passes that value to its MoneyHandler object sending the add message to it

play

- Tell the MoneyHandler object to deduct 0.25 to pay for the spin (and print a message showing the amount paid)
- Send the spin message to the PlayingField, which will return a number. Send that value to the MoneyHandler's add method
- Print out the current balance using the MoneyHandler's toString method

quit

- Send *dispense* to the MoneyHandler.
- Ends execution of the game

Sample Output:

>java Launcher

How much money would you like to insert? 50

Spin/Quit (S/Q) : s

Paid 0.25 to spin....

Spun Donald-Mickey-Mickey

Won 50 cents

You have a \$ 50.25 balance

Spin/Quit (S/Q) : s

Paid 0.25 to spin....

Spun Mickey-Minnie-Mickey

Won 50 cents

You have a \$ 50.50 balance

Spin/Quit (S/Q) : s

Paid 0.25 to spin....

Spun Daisy-Mickey-Daisy

Won 50 cents

You have a \$ 50.75 balance

Spin/Quit (S/Q) : s

Paid 0.25 to spin....

Spun Daisy-Daisy-Donald

Won 50 cents

You have a \$ 51.00 balance

Spin/Quit (S/Q) : s

Paid 0.25 to spin....

Spun Mickey-Daisy-Minnie

No prize

You have a \$ 50.75 balance

Spin/Quit (S/Q) : s

Paid 0.25 to spin....

Spun Donald-Minnie-Donald

Won 50 cents

You have a \$ 51.00 balance

Spin/Quit (S/Q) : q

Dispensed \$ 51.00

