**David Hazall-Farrell**

**CS-132-Lab 8**

**6 hours**

**I was not a fan of this lab. It started smoothly, but when the Bank Account class was reintroduced and we had to extend it, it got very annoying. The lab could’ve been easier if it just wasn’t in the lab.**

Dice dice = **new** Dice(100);

// Create an ArrayList to hold the 20 random integers

ArrayList<Integer> randomValues = **new** ArrayList<>();

// Loop to generate and store 20 random values in the ArrayList

**for** (**int** i = 0; i < 20; i++) {

**int** roll = dice.roll(); // Using Dice class's roll method

randomValues.add(roll);

}

// Loop to display the values stored in the ArrayList, one per line

**for** (Integer value : randomValues) {

System.***out***.println(value);

}

}

4.

**for** (**int** i = 0; i < 20; i++) {

**int** roll = dice.roll(); // Using Dice class's roll method

**int** pos = *findPosition*(roll, randomValues); // Find the position to insert

randomValues.add(pos, roll); // Insert the value in the correct position

}

// Loop to display the values stored in the ArrayList, one per line

**for** (Integer value : randomValues) {

System.***out***.println(value);

}

}

Output:

4

6

7

20

21

24

31

33

35

36

44

48

48

53

58

65

72

79

80

96

5.

Nine days old.

Nine days old.

Pease porridge cold,

Pease porridge hot,

Pease porridge in the pot

Some like it cold,

Some like it hot,

Some like it in the pot

6.

eption in thread "main" java.lang.Error: Unresolved compilation problem:

The method findPosition(Object, ArrayList<Object>) in the type OrderedList is not applicable for the arguments (int, ArrayList<Integer>)

at OrderedList.main(OrderedList.java:16)

It doesn’t work since the findposition class can’t connect to the OrderedList method since it is now an object.

7.

6

9

13

17

17

18

22

24

26

28

28

32

41

53

54

63

69

93

95

100

8-9.

**public** **class** BankAccount **implements** Comparable<BankAccount> {

//Balance

//the account number

//the name of the customer

**private** **double** balance;

**private** String customerName;

**private** String accountNumber;

**public** BankAccount(String custName, **double** bal, String acctNum) {

**this**.customerName = custName;

**this**.balance = bal;

**this**.accountNumber = acctNum;

}

**public** String toString() {

**return** "Account Num: " + **this**.getAccountID() + ", " + **this**.getCustomerName() + ", " + **this**.getBalance();

}

**public** String getAccountID() {

**return** accountNumber;

}

**public** String getCustomerName() {

**return** customerName;

}

**public** **double** getBalance() {

**return** balance;

}

**public** **void** setBalance(**double** balance) {

**this**.balance = balance;

}

**public** **void** deposit(**double** amountToDeposit) {

**this**.balance = **this**.balance + amountToDeposit;

}

**public** **void** withdraw(**double** amountToWithdraw) {

**if**(**this**.balance - amountToWithdraw < 0) {

**throw** **new** IllegalArgumentException();

}

**this**.balance = **this**.balance - amountToWithdraw;

}

@Override

**public** **int** compareTo(BankAccount other) {

// Compare based on account number (alphabetically)

**return** **this**.accountNumber.compareTo(other.accountNumber);

}

}