First I dove into the PiggyBank class and used old recycled code from last year only changing up the public class to implement serializable for the file. This is also where all the math and logic is applied for the manipulation of data in the program.

```
public class PiggyBank implements Serializable {



        double totalAmount = 0;
        DecimalFormat format = new DecimalFormat("0.00");
        public PiggyBank() {
        totalAmount = 0;
        }
        public void addPenny () {
        totalAmount += 0.01;
        }
        public void addNickel () {
        totalAmount += 0.05;
        }
        public void addDime () {
        totalAmount += 0.10;
        }
        public void addQuarter () {
        totalAmount += 0.25;
        }

        public void removeM (double remove) {

        if (remove < totalAmount && remove > 0) {
        totalAmount -= remove;
        }

        else if (totalAmount < remove || remove < 0) {
        System.out.println("Try again! The amount you wish to take out is more than you currently have!");
        }

    }


        public String toString() {
        String call = ("The total amount of money in the users account is: " + format.format(totalAmount) + "$");
        return call;
    }
}
```

Next I started by asking the user if the program may load the previous file (or save). This will load the same file back however if I had implemented a system to take the file directory from the user then accounts could be made for each individual who may use the program instead of using just one file to store all user data. After this the program will read the file and takes account of the data the file had previously stored.

```
System.out.println("Do you wish to load the previous file? (Y/N)");
user = input.nextLine();

if(user.toLowerCase().equals("y"))
{
    try{
        textFile = new File("C:\\Users\\77132002\\git\\CS30P3F2024\\Chapter11\\src\\Masteries\\pgBank.txt");
        FileInputStream in = new FileInputStream(textFile);
        ObjectInputStream readStu = new ObjectInputStream(in);


        userAccount = (PiggyBank) readStu.readObject();




        NumberFormat money = NumberFormat.getCurrencyInstance();

        in.close();
        readStu.close();
    }

    catch (FileNotFoundException e)
    {
        System.out.println("File does not exist or could not be found.");
        System.err.println("FileNotFoundException: " + e.getMessage());
    }
    catch (IOException e)
    {
        System.out.println("Problem ready file");
        System.err.println("IOException: " + e.getMessage());
    }
    catch (ClassNotFoundException e)
    {
        System.out.println("Class does not exist or could not be found.");
        System.err.println("ClassNotFoundException: " + e.getMessage());
    }
```

The next bit is also just recycled code from last year in which is used from taking user inputs and data.

```
do {
System.out.println("1. Show total in bank.");
System.out.println("2. Add a penny");
System.out.println("3. Add a nickel");
System.out.println("4. Add a dime");
System.out.println("5. Add a quarter");
System.out.println("6. Take money out of bank");
System.out.println("Enter 0 " + " to quit");
System.out.println("Enter your choice: ");

data = input.nextDouble();


if (data == 1)
{
System.out.println(userAccount);
}

else if (data == 2)
{
userAccount.addPenny();
System.out.println("A penny has been added to your account!");
}

else if (data == 3)
{
userAccount.addNickel();
System.out.println("A nickel has been added to your account!");
}

else if (data == 4)
{
userAccount.addDime();
System.out.println("A dime has been added to your account!");
}

else if (data == 5)
{
userAccount.addQuarter();
System.out.println("A quarter has been added to your account!");
}

else if (data == 6)
{
System.out.println("How much would you like to take out?: ");
withdraw = input.nextDouble();
userAccount.removeM(withdraw);
}

else if (data == 0) {
System.out.println(" ");
}



    } while (data != 0);

}
```

Lastly I use another try catch statement to Write all data the user had provided into the file

```
try {
    textFile = new File("C:\\Users\\77132002\\git\\CS30P3F2024\\Chapter11\\src\\Masteries\\pgBank.txt");

    FileOutputStream out = new FileOutputStream(textFile);
    ObjectOutputStream writeStu = new ObjectOutputStream(out);

    writeStu.writeObject(userAccount);



    writeStu.close();
    System.out.println("Data written to file. ");
}


catch (FileNotFoundException e)
{
    System.out.println("File does not exist or could not be found.");
    System.err.println("FileNotFoundException: " + e.getMessage());
}
catch (IOException e)
{
    System.out.println("Problem ready file");
    System.err.println("IOException: " + e.getMessage());
}
```