

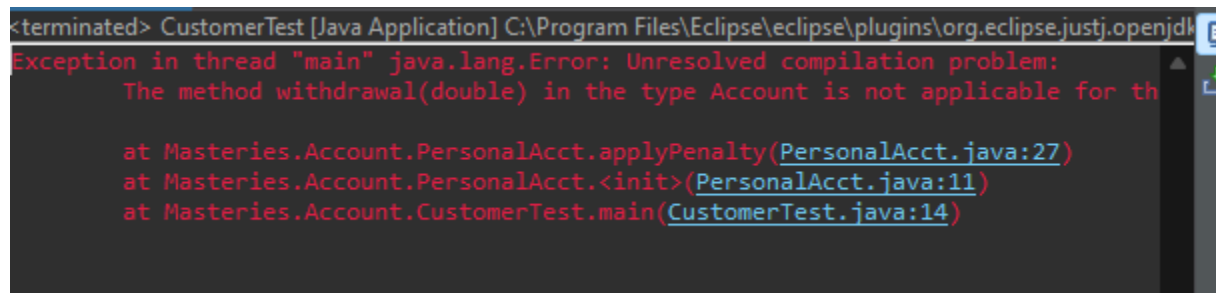
Credit Name: Chapter 8

Assignment Name: Account Mastery

Name: Grayson Ardron

Error Log

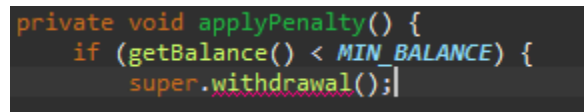
The error that was thrown was this shown in the picture below

A screenshot of the Eclipse IDE's error log window. The title bar reads "<terminated> CustomerTest [Java Application] C:\Program Files\Eclipse\eclipse\plugins\org.eclipse.justj.openjdk". The log contains a red message: "Exception in thread \"main\" java.lang.Error: Unresolved compilation problem: The method withdrawal(double) in the type Account is not applicable for th". Below this, the stack trace is shown in red: "at Masteries.Account.PersonalAcct.applyPenalty(PersonalAcct.java:27)", "at Masteries.Account.PersonalAcct.<init>(PersonalAcct.java:11)", and "at Masteries.Account.CustomerTest.main(CustomerTest.java:14)".

```
<terminated> CustomerTest [Java Application] C:\Program Files\Eclipse\eclipse\plugins\org.eclipse.justj.openjdk
Exception in thread "main" java.lang.Error: Unresolved compilation problem:
    The method withdrawal(double) in the type Account is not applicable for th

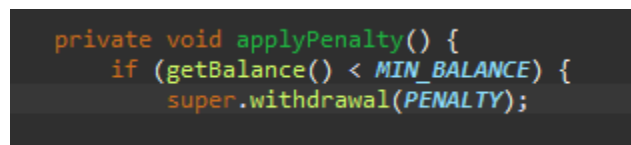
    at Masteries.Account.PersonalAcct.applyPenalty(PersonalAcct.java:27)
    at Masteries.Account.PersonalAcct.<init>(PersonalAcct.java:11)
    at Masteries.Account.CustomerTest.main(CustomerTest.java:14)
```

And the issue I found it out to be was that i hadn't added the PENALTY variable into the the super method

A screenshot of a code editor showing the applyPenalty method. The code is: "private void applyPenalty() { if (getBalance() < MIN_BALANCE) { super.withdrawal(); } }". The word "withdrawal" is underlined with a red squiggly line, indicating a compilation error.

```
private void applyPenalty() {
    if (getBalance() < MIN_BALANCE) {
        super.withdrawal();
    }
}
```

It was an easy fix as shown below

A screenshot of a code editor showing the corrected applyPenalty method. The code is: "private void applyPenalty() { if (getBalance() < MIN_BALANCE) { super.withdrawal(PENALTY); } }". The word "PENALTY" is in blue, indicating it is a recognized variable.

```
private void applyPenalty() {
    if (getBalance() < MIN_BALANCE) {
        super.withdrawal(PENALTY);
    }
}
```