**Bug1**  
  
Bug Report: https://bugzilla-dev.allizom.org/show\_bug.cgi?id=1633610

ClassName: Class\_ //path🡪 JavaSource/org/unitime/timetable/model/Class\_.java

Issue: Division by zero in the "effectiveTeachingLoad" method.

Description: The "effectiveTeachingLoad" method calculates the effective teaching load by dividing the total load by the number of instructors. However, there is no check to ensure that the number of instructors is not zero before performing the division operation. This could result in a division by zero error, leading to unexpected behavior or program termination.

**Bug2**  
  
Bug Report: https://bugzilla-dev.allizom.org/show\_bug.cgi?id=1633609

ClassName:ExamDetailAction.  
//path🡪JavaSource/org/unitime/timetable/action/ExamDetailAction.java

Issue: String and boxed types comparison using reference equality instead of equals() method.

Description: The code is comparing two instances of the String and boxed types using reference equality (== or !=) instead of using the equals() method. This can lead to unexpected behavior as it compares the memory location of the two instances rather than their values. This can result in the program not behaving as expected.

**Bug 3**  
Bug Report: <https://bugzilla-dev.allizom.org/show_bug.cgi?id=1633663>

ClassName: ClassesAction

Path: JavaSource/org/unitime/timetable/action/ClassesAction.java

Issue: The code contains a logical error in the if statement, which may cause unexpected behavior.

Description:

In the code snippet if (subject != null subject != null), the same condition subject != null is used on both sides of the operator. This means that the condition will always be true, regardless of the value of subject. This may cause the subsequent code to execute incorrectly, as the intended behavior may not be properly defined.

**Bug 4**

Bug Report: https://bugzilla-dev.allizom.org/show\_bug.cgi?id=1633664

ClassName: ReservationExport

Path: JavaSource/org/unitime/timetable/dataexchange/ReservationExport.java

Issue: The code contains a logical error where a duplicate condition is being checked in a sequence of if/else if statements. This causes a branch of the code to be unreachable and can cause unexpected behavior or errors in the program.

Description: In the given code, a duplicate condition is being checked in a sequence of if/else if statements. Specifically, the condition reservation instanceof OverrideReservation is being checked twice, which causes a portion of the code to be unreachable and may lead to incorrect results or runtime errors.