

CS6308 –JAVA PROGRAMMING LAB

Built in Exceptions

DATE: 23.3.23

Question A

1. The Employee constructor throws **NullPointerException** if name and designation is initialised with null.
2. The PermanentEmployee constructor throws **IllegalArgumentException** if the basicPay and gradePay are initialised with arguments with negative or zero value
3. The Salesperson constructor throws **IllegalArgumentException** if the basicSalary is initialised with arguments with negative or zero value
4. In the Salesperson class, addSale(double amount) throws **IllegalArgumentException** if the argument passed Amount is negative or zero value
5. Create an array of objects of Employee initialised with objects of PermanentEmployee and Salesperson, throw **ArrayIndexOutOfBoundsException** when printing the details of the array objects
6. The test employee throws **InputMismatchException** when the input obtained from user mismatches with the required type
7. Include **finally** block wherever necessary

Question B

1. The **Account** constructor throws **NullPointerException** if accname and contactAddress is initialised with null. It also throws **IllegalArgumentException** if the accno and balance is initialised with zero or negative value.
2. withdraw() throws **Exception** if the amount to be withdrawn is greater than balance and check before deducting amount from balance whether the balance after deduction is less than 500 if so throw Exception.
3. Withdraw in **CheckingsAccount** should throw **Exception** if balance after withdraw is less than 1000.
4. getTotalTransactions() in **SavingsAccount** should throw **Exception** if the number of transactions exceeds 10.
5. Create an array of objects of Account initialised with objects of **CheckingsAccount** and **SavingsAccount** , throw **ArrayIndexOutOfBoundsException** when printing the details of the array objects
6. The **TestAccount** throws **InputMismatchException** when the input obtained from user mismatches with the required type
7. Include **finally** block wherever necessary

Question C

1. The Course constructor throws **NullPointerException** if courseName, grade and type is initialised with null. It also throws **IllegalArgumentException** if the credits and contactHrs is initialised with zero or negative value.
2. calculateGrade(double) throws **IllegalArgumentException** if the argument passed is negative value.
3. Student constructor **NullPointerException** if name is initialised with null. It also throws **IllegalArgumentException** if the cgpa and rollNo is initialised with zero or negative value.
4. enrollCourse(Course) throws Exception if the maxCredits exceeds 24 before adding course to courselist
5. findCourse(Course) throws **IllegalArgumentException** if the argument passed is null
6. modifyCredits(int) throws Exception if the argument passed is negative or zero
7. dropCourse(Course) throws **IllegalArgumentException** if the argument passed is null
8. PartTimeStudent constructor throws **IllegalArgumentException** if the maxCourses is initialised with zero or negative value.
8. **TestStudent** throws **InputMismatchException** when the input obtained from user mismatches with the required type
9. Include **finally** block wherever necessary
10. cal_CGPA() throws ArithmeticException if the denominator is zero

Question D

1. Player constructor throws **IllegalArgumentException** if the id and balance is initialised with zero or negative value.
2. PremiumPlayer constructor throws **IllegalArgumentException** is initialised with zero or negative value.
3. **TestPlayer** throws **InputMismatchException** when the input obtained from user mismatches with the required type
4. Include **finally** block wherever necessary
5. Create an array of objects of **Players** initialised with objects of PremiumPlayer throw **ArrayIndexOutOfBoundsException** when printing the details of the array objects