

# Core Java Assignments

1. WAP to take marks of 3 subjects from the User, compute total marks secured and percentage.

2. In the above program, Compute Grade of the Student based on following criteria:

percent > 75: Grade A

percent > 60: Grade B

else : Grade C

3. WAP to implement following Interface for implementing banking operations.

```
interface Deposit{
```

```
    void deposit(Customer customer, double amount);
```

```
}
```

```
interface Withdrawal{
```

```
    double limit=50000;
```

```
    void withdraw(Customer customer, double amount)
```

```
}
```

Create Service classes(DepositService and WithdrawalService) to implement Deposit and Withdrawal interfaces respectively.

Call these service classes from App class and perform random deposits and withdrawals on at least 2 customers.

Customer class

id <int / Integer>

name <String>

balance <Double / double>

Note: Use Encapsulation to design Customer class.

4. Create 3 Employee Objects having following details

id:1 name: harry potter city: London salary: 85000	id:2 name: ronald weasley city: surrey salary: 75000	id:3 name: hermione granger city: london salary: 95000
---	---	---

Save these object in List and perform following operations: [Use either Comparable or Comparator Interface]

a. Sort as per salary ASC order

b. Sort as per salary DESC order

5. In the above case study, filter employees based on following criteria:

- a. Display employees having salary<80000
- b. Display employees living in city 'london'

**6.** In case study 3 above, perform following validations using self defined exceptions.

**InsufficientFundsException:**

if amount > balance of the customer, throw this exception with the message "Insufficient Funds"

**OverTheLimitException:**

if amount > 50000 during withdrawal, throw this exception with the message "Limit 50000 Exceeded"

Note: Both the exceptions should be checked exceptions.

**7.** Login System using Map.

Save 5 username/passwords in HashMap with username as key and password as value.

Take username/password as Input from the User and check if they are valid against the entries of HashMap.

**8.** Case Study: Menu Driven Program using JDBC API

Create a table product(id,name,price,description) in MySql DB.

Case 1:

Write a program to Insert the records in the table using JDBC API. Note: Taking input from User.

Case 2: Display all products from the DB

Case 3: Delete product based on id

Case 0: exit