# Phase-1-Final Assessment

## Camera Rental Application
## (Course End Project-1)

## Source Code:

### A)Camera.java

```
package camerarentalapplication;
public class Camera {
	private int CameraID;
	private String brand;
	private String model;
	private double pricePerDay;
	private String status;

	public Camera(int CameraID, String brand, String model, double pricePerDay, String status)
{
		this.CameraID=CameraID;
		this.brand=brand;
		this.model=model;
		this.pricePerDay=pricePerDay;
		this.status=status;
	}
	public int getCameraID() {
		return CameraID;
	}
	public String getBrand() {
		return brand;
	}
	public String getModel() {
		return model;
	}
	public double getPricePerDay() {
		return pricePerDay;
	}
	public String getStatus() {
		return status;
	}

	public void setStatus(String status) {
		this.status=status;
	}

	public String toString() {
		return String.format("%-10s  %-12s %-12s  %-16s %s", CameraID, brand, model,
pricePerDay, status);
	}
```

```
}
```

## B)Wallet.java

```
package camerarentalapplication;
public class Wallet {
	private double balance;

	public Wallet() {
		this.balance=0.0;
	}
	public double getBalance() {
		return balance;
	}

	public void deposite(double amount) {
		if(amount>0) {
			balance+=amount;
			System.out.println("Your wallet has been credited with INR" +amount);
			System.out.println("Your current wallet balance is INR" +balance);
		}else {
			System.out.println("Invalid amount.Deposite amount should be greater than
zero.");
			}
	}
	public void withdraw(double amount) {
		if(amount>0 && amount<=balance) {
			balance-=amount;

			System.out.println("INR" +amount+ "has been debited from your wallet.");
			System.out.println("Your current wallet balance is INR" +balance);
		}else {
			System.out.println("Insufficient balance or invalid amount.");
		}
	}
}
```

## C)CameraRental.java

```
package camerarentalapplication;
import java.util.ArrayList;
import java.util.List;
import java.util.Scanner;
public class CameraRental {

	private static List<Camera> cameraList=new ArrayList<>();
	private static Wallet wallet=new Wallet();

	public static void main(String[] args) {
		Scanner scanner=new Scanner(System.in);
```

```
        System.out.println("+---------------------------------------------------------+");
        System.out.println("        WELCOME TO CAMERA RENTAL APP                      ");
        System.out.println("+---------------------------------------------------------+");

        System.out.print("Hi, PLEASE LOGIN TO CONTINUE");


        System.out.print("\nUSERNAME: ");
        String username=scanner.nextLine();
        System.out.print("PASSWORD: ");
        String password=scanner.nextLine();


        if (login(username, password)) {
                handleMainMenu(scanner);
        }else {
                System.out.println("Invalid username or password.  Exiting the application.");

        }
}

private static boolean login(String username, String password) {

        return username.equals("Varshitha") && password.equals("password");
}

private static void handleMainMenu(Scanner scanner) {
        int choice;
         do {
                System.out.println("\nOPTIONS TO BE DISPLAYED:");
                System.out.println("1.MY CAMERA");
                System.out.println("2.RENT A CAMERA");
                System.out.println("3.VIEW ALL CAMERAS");
                System.out.println("4.MY WALLET");
                System.out.println("5.EXIT");
                System.out.println("\nEnter your choice:");

                choice=scanner.nextInt();
                scanner.nextLine();


                switch(choice) {

                case 1:
                        handleMyCamera(scanner);
                        break;

                case 2:
                        handleRentCamera(scanner);
                        break;
```

```
                case 3:
                        handleViewAllCamrea();
                        break;

                case 4:
                        handleMyWallet(scanner,wallet);
                        break;

                case 5:
                        System.out.println("Exiting the application. Goodbye!");
                        break;
                        default:

                                System.out.println("Invalid choice. Please try again.");
                }
        }while(choice!=5);
}


private static void handleMyCamera(Scanner scanner) {
        int choice;
        do {

                System.out.println("\n1.ADD");
                System.out.println("2.REMOVE");
                System.out.println("3.VIEW MY CAMERAS");
                System.out.println("4.GO TO PREVIOUS MENU");
                System.out.println("\nEnter your choice");

                choice=scanner.nextInt();
                scanner.nextLine();


                switch(choice) {
                case 1:
                        handleAddCamera(scanner);
                        break;

                case 2:
                handleRemoveCamera(scanner);
                break;

                case 3:
                        handleViewMyCamera(scanner);
                        break;

                case 4:
                        System.out.println("Going back to the previous menu.");
                        break;
                        default:

                                System.out.println("Invalid choice. Please try again.");
```

```
                }
            }while(choice!=4);
        }



        private static void handleAddCamera(Scanner scanner) {
            System.out.println("\nADD CAMERA:");
            System.out.println("+----------------------------------------------------+");
            System.out.println("       WELCOME TO CAMERA RENTAL APP                 ");

            System.out.println("ENTER THE CAMERA BRAND");
            String brand=scanner.nextLine();

            System.out.println("ENTER THE MODEL");
            String model=scanner.nextLine();

            System.out.println("ENTER THE PER DAY PRICE(INR)");
            double pricePerDay=scanner.nextDouble();
            scanner.nextLine();


            int cameraID=generateUniqueID();
            Camera newCamera=new Camera(cameraID,brand,model,pricePerDay, "Available");
            cameraList.add(newCamera);

            System.out.println("YOUR CAMERA HAS BEEN SUCCESSFULLY ADDED  TO THE
LIST.");
        }


        private static void handleRemoveCamera(Scanner scanner) {
            System.out.println("\nREMOVE CAMERA:");

            System.out.println("+----------------------------------------------------+");
            System.out.println("       WELCOME TO CAMERA RENTAL APP                 ");
            System.out.println("+----------------------------------------------------+");

            System.out.println("ENTER THE CAMERA ID TO REMOVE:");
            int cameraId=scanner.nextInt();
            scanner.nextLine();

            boolean removed=false;
            for(Camera camera:cameraList) {
                if(camera.getCameraID()==cameraId) {
                    cameraList.remove(camera);
                    removed=true;
                    break;
                }
            }
```

```
                if(removed) {
                        System.out.println("CAMERA REMOVED SUCCESSFULLY.");
                }else {
                        System.out.println("CAMERA NOT FOUND.");
                }
        }


        private static void handleViewMyCamera(Scanner scanner) {
                System.out.println("\nVIEW MY CAMERAS:");

                System.out.println("+-----------------------------------------------+");
                System.out.println("      WELCOME TO CAMERA RENTAL APP              ");
                System.out.println("+-----------------------------------------------+");

                if(cameraList.isEmpty()) {
                        System.out.println("No cameras found.");
                }else {
                        System.out.println("MY CAMERAS:");

                        System.out.println("+-------------------------------------------------------------+");
                        System.out.println("CAMERA   ID    BRAND    MODEL    PRICE (PER DAY)
STATUS ");
                        System.out.println("+-------------------------------------------------------------");

                        for(Camera camera:cameraList) {
                                System.out.println(camera);
                        }

                        System.out.println("+-------------------------------------------------------------+");
                }
        }


        private static void handleRentCamera(Scanner scanner) {
                System.out.println("\nRENT A CAMERA:");

                System.out.println("+-----------------------------------------------------------------+");
                System.out.println("               WELCOME TO CAMERA RENTAL APP
");
                System.out.println("+-----------------------------------------------------------------+");

                handleViewAllCamrea();

                System.out.println("ENTER THE CAMERA ID TO RENT:");
                int cameraId=scanner.nextInt();
                scanner.nextLine();

                Camera rentedCamera=null;
                for(Camera camera:cameraList) {
                        if(camera.getCameraID()==cameraId) {
                                rentedCamera=camera;
```

```
                        break;
                    }
                }

                if(rentedCamera!=null) {
                    if
                    (rentedCamera.getStatus().equals("Available")) {
                        System.out.println("ENTER THE NUMBER OF DAYS TO RENT:");
                        int numberOfDays=scanner.nextInt();
                        scanner.nextLine();

                        double totalPrice=rentedCamera.getPricePerDay() * numberOfDays;
                        if(wallet.getBalance()>=totalPrice) {
                            wallet.withdraw(totalPrice);
                            rentedCamera.setStatus("Rented");
                            System.out.println("CAMERA RENTED SUCCESSFULLY.");
                        }else {
                            System.out.println("INSUFFICIENT BALANCE IN YOUR
WALLET.");
                        }
                    }else {
                        System.out.println("CAMERA IS NOT AVAILABLE FOR RENT.");
                    }
                }else {
                    System.out.println("CAMERA NOT FOUND.");
                }
            }

            private static void handleViewAllCamrea() {
                System.out.println("\nVIEW ALL CAMERAS:");
                System.out.println("+-----------------------------------------+");
                System.out.println("     WELCOME TO CAMERA RENTAL APP            ");
                System.out.println("+-----------------------------------------+");


                if(cameraList.isEmpty()) {
                    System.out.println("No Cameras found.");
                }else {
                    System.out.println("ALL CAMERAS:");

                    System.out.println("+-------------------------------------------------------+");
                    System.out.println("CAMERA  ID  BRAND  MODEL     PRICE(PER DAY)
STATUS ");
                    System.out.println("+-------------------------------------------------------+");

                    for(Camera camera:cameraList) {
                        System.out.println(camera);
                    }

                    System.out.println("+-------------------------------------------------------+");
                }
            }
```

```
	private static void handleMyWallet(Scanner scanner, Wallet wallet) {
		System.out.println("\nMY WALLET:");

		System.out.println("+---------------------------------------+");
		System.out.println("       WELCOME TO CAMERA RENTAL APP        ");
		System.out.println("+---------------------------------------+\n");

		System.out.println("YOUR CURRENT WALLET BALANCE IS INR" +
wallet.getBalance());

		System.out.println("SELECT AN OPTION:");
		System.out.println("1.Deposite money");
		System.out.println("2.Go back to the previous menu");


		System.out.println("Enter your choices: ");
		int choice=scanner.nextInt();
		scanner.nextLine();


		switch(choice) {

		case 1:
			System.out.println("ENTER THE AMOUNT TO DEPOSITE(INR):");
			double depositeAmount=scanner.nextDouble();
			scanner.nextLine();
			wallet.deposite(depositeAmount);
			break;

		case 2:
			System.out.println("Going back to the previous menu.");
			System.out.println("Invalid choice.Please try again.");
			break;
		}
	}


	private static int generateUniqueID() {

		int maxId=0;
		for(Camera camera:cameraList) {
			if(camera.getCameraID()>maxId) {
				maxId=camera.getCameraID();
			}
		}return maxId+1;
	}
}
```