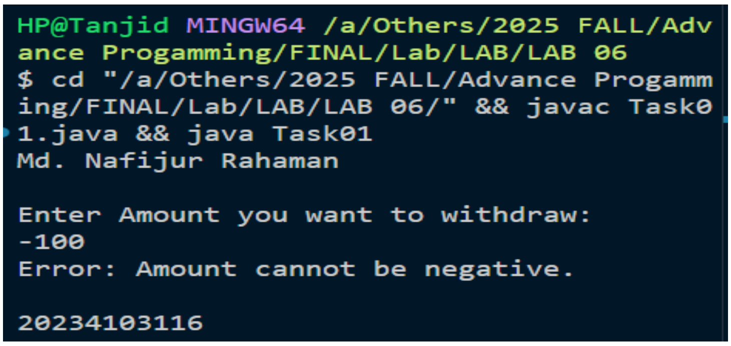
Task01:

```
import java.util.Scanner;
class Account {
    String accountNumber;
    double balance;

    Account(String accountNumber, double balance) {
        this.accountNumber = accountNumber;
        this.balance = balance;
    }

    void withdraw(double amount) throws IllegalArgumentException {

        if (amount < 0) {
            throw new IllegalArgumentException("Amount cannot be negative.");
        }
        else if (amount > this.balance) {
            throw new IllegalArgumentException("Insufficient balance.");
        }
        else {
            this.balance -= amount;
            System.out.println("Withdrawal successful.");
            System.out.println("Remaining Balance: " + this.balance);
        }
    }
}

public class Task01 {
    public static void main(String[] args) {

        System.out.println("Md. Nafijur Rahaman\\n");

        Account account = new Account("123456", 1000);

        System.out.println("Enter Amount you want to withdraw: ");
        Scanner in = new Scanner(System.in);
        double amount = in.nextDouble();

        try {
            account.withdraw(amount);
        } catch (IllegalArgumentException e) {
            System.out.println("Error: " + e.getMessage());
        }

        System.out.println("\\n20234103116");
    }}
```

Output:

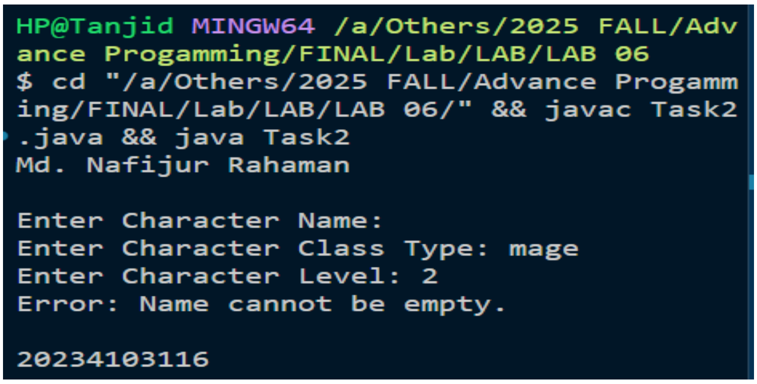


```
HP@Tanjid MINGW64 /a/Others/2025 FALL/Adv
ance Progamming/FINAL/Lab/LAB/LAB 06
$ cd "/a/Others/2025 FALL/Advance Progamm
ing/FINAL/Lab/LAB/LAB 06/" && javac Task0
1.java && java Task01
Md. Nafijur Rahaman

Enter Amount you want to withdraw:
-100
Error: Amount cannot be negative.

20234103116
```

Task02

```
import java.util.Scanner;

class Character {
    String name, class_type;
    int level;

    void CreateCharacter(String name, String class_type, int level) {
        this.name = name;
        this.class_type = class_type;
        this.level = level;
    }

    String getName() throws IllegalArgumentException {
        if (this.name.isEmpty()) {
            throw new IllegalArgumentException("Name cannot be empty.");
        }
        return this.name;
    }

    String getClassType() throws IllegalArgumentException {
        if (!class_type.equalsIgnoreCase("warrior") && !class_type.equalsIgnoreCase("mage"))
{
            throw new IllegalArgumentException("Invalid class type. Allowed: warrior or mage.");
        }
        return this.class_type;
    }

    int getLevel() throws IllegalArgumentException {
        if (this.level < 1 || this.level > 100) {
            throw new IllegalArgumentException("Level must be between 1 and 100.");
        }
        return this.level;
    }
}

public class Task2 {
    public static void main(String[] args) {

        System.out.println("Md. Nafijur Rahaman\\n");

        Scanner in = new Scanner(System.in);

        Character character = new Character();
        System.out.print("Enter Character Name: ");
        String name = in.nextLine();
        System.out.print("Enter Character Class Type: ");
        String class_type = in.nextLine();
        System.out.print("Enter Character Level: ");
        int level = in.nextInt();

        character.CreateCharacter(name, class_type, level);

        try {
            System.out.println("Character Name: " + character.getName());
            System.out.println("Character Class Type: " + character.getClassType());
            System.out.println("Character Level: " + character.getLevel());
        } catch (IllegalArgumentException e) {
            System.out.println("Error: " + e.getMessage());
        }

        System.out.println("\\n20234103116");

    }}
```

Output:

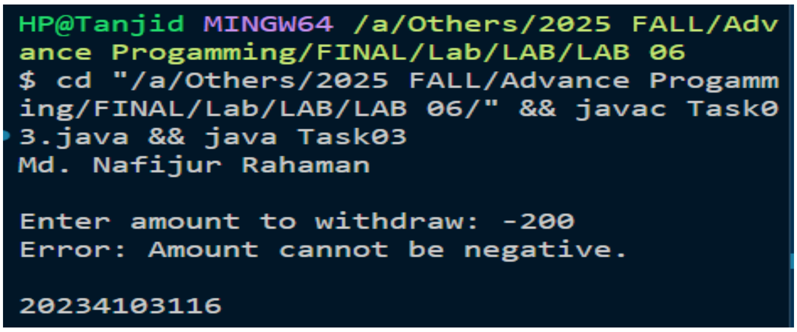


```
HP@Tanjid MINGW64 /a/Others/2025 FALL/Advance Progamming/FINAL/Lab/LAB/LAB 06
$ cd "/a/Others/2025 FALL/Advance Progamming/FINAL/Lab/LAB/LAB 06/" && javac Task2.java && java Task2
Md. Nafijur Rahaman

Enter Character Name:
Enter Character Class Type: mage
Enter Character Level: 2
Error: Name cannot be empty.

20234103116
```

Task 03:

```
import java.util.Scanner;

class InsufficientFundsException extends Exception {
    public InsufficientFundsException(String message) {
        super(message);
    }
}

class NegativeAmountException extends Exception {
    public NegativeAmountException(String message) {
        super(message);
    }
}

class BankAccount {
    private double balance;

    BankAccount(double balance) {
        this.balance = balance;
    }

    void withdraw(double amount) throws InsufficientFundsException,
NegativeAmountException {

        if (amount < 0) {
            throw new NegativeAmountException("Amount cannot be negative.");
        }

        if (amount > balance) {
            throw new InsufficientFundsException("Insufficient balance. Available: " + balance);
        }

        balance -= amount;
        System.out.println("Withdrawal successful. Remaining balance: " + balance);
    }
}

public class Task03 {
    public static void main(String[] args) {
        System.out.println("Md. Nafijur Rahaman\\n");

        BankAccount account = new BankAccount(1000);
        Scanner sc = new Scanner(System.in);

        System.out.print("Enter amount to withdraw: ");
        double amount = sc.nextDouble();
        try {
            account.withdraw(amount);
        } catch (NegativeAmountException e) {
            System.out.println("Error: " + e.getMessage());
        } catch (InsufficientFundsException e) {
            System.out.println("Error: " + e.getMessage());
        }
        System.out.println("\\n20234103116");

    }
}
```

Output:

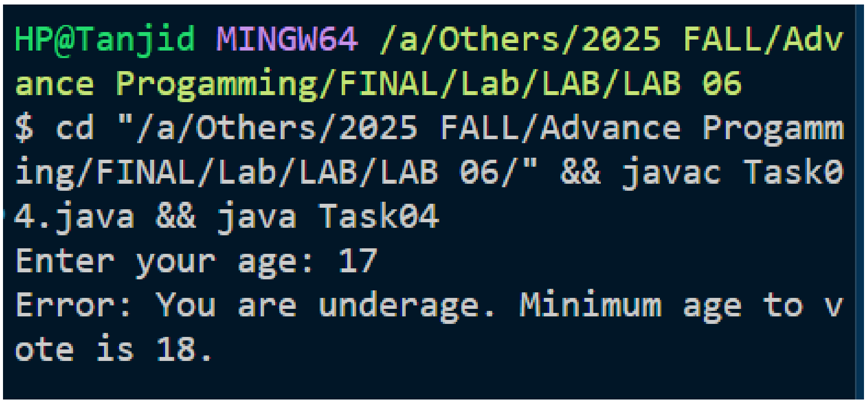


```
HP@Tanjid MINGW64 /a/Others/2025 FALL/Adv
ance Progamming/FINAL/Lab/LAB/LAB 06
$ cd "/a/Others/2025 FALL/Advance Progamm
ing/FINAL/Lab/LAB/LAB 06/" && javac Task0
3.java && java Task03
Md. Nafijur Rahaman

Enter amount to withdraw: -200
Error: Amount cannot be negative.

20234103116
```

Task04:

```
import java.util.Scanner;

class UnderageException extends Exception {
    public UnderageException(String message) {
        super(message);
    }
}

public class Task04 {
    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);
        System.out.print("Enter your age: ");
        int age = sc.nextInt();

        try {
            checkEligibility(age);
            System.out.println("You are eligible to vote in the USA.");
        } catch (UnderageException e) {
            System.out.println("Error: " + e.getMessage());
        }

        sc.close();
    }

    static void checkEligibility(int age) throws UnderageException {
        if (age < 18) {
            throw new UnderageException("You are underage. Minimum age to vote is 18.");
        }
    }
}
```

Output:



```
HP@Tanjid MINGW64 /a/Others/2025 FALL/Adv
ance Progamming/FINAL/Lab/LAB/LAB 06
$ cd "/a/Others/2025 FALL/Advance Progamm
ing/FINAL/Lab/LAB/LAB 06/" && javac Task0
4.java && java Task04
Enter your age: 17
Error: You are underage. Minimum age to v
ote is 18.
```