ASSIGNMENT -2

Java Classes

a) Define a class Time having four private data members; hour, min, and sec. The class must have folllowing methods:

→ A default constructor

→ A parameterized constructor

→ A displayTime() method to display the time in HH:MM:SS format. → A setTime(byte,byte,byte) to set the values of hour, min, and sec. → An addTime(Time,Time) method to add two Time objects passes as arguments and store the result in the object that has called the method.

→ A subtractTime(Time,Time) method : to subtract two Time objects passes as arguments and store the result in the object that has called the method.

→ toSeconds() method to convert the time into number of seconds.

→ tohours() method to convert the time into total number of hours

CODE :-

```
import java.util.Scanner;

class Time {

   private int hour, min, sec;

   Time() {

      hour = 0;

      min = 0;

      sec = 0;

   }

   Time(int h, int m, int s) {

      this.hour = h;
```

```
    this.min = m;
    this.sec = s;
}
void displayTime() {
    System.out.print("Time : ");
    if(hour < 10){
        System.out.print("0");
    }
    System.out.print(hour + ":");
    if(min < 10){
        System.out.print("0");
    }
    System.out.print(min + ":");
    if(sec < 10){
        System.out.print("0");
    }
    System.out.println(sec);
}
void setTime(int h, int m, int s) {
    this.hour = h;
    this.min = m;
    this.sec = s;
}
Time addTime(Time t1, Time t2) {
    int c1 = 0;
    int c2 = 0;
    int s = t1.sec + t2.sec;
    if (s >= 60) {
        c1++;
        s = (s % 60);
    }
    int m = t1.min + t2.min + c1;
    if (m >= 60) {
        c2++;
        m = (m % 60);
    }
    int h = t1.hour + t2.hour + c2;
    h = h % 24;
    return new Time(h, m, s);
}
Time subtractTime(Time t1, Time t2) {
    int s = t1.sec - t2.sec;
    int m = t1.min - t2.min;
    int h = t1.hour - t2.hour;
    if (s < 0) {
        s += 60;
        m--;
    }
    if (m < 0) {
        m += 60;
        h--;
    }
```

```
        if (h < 0) {
            h += 24;
        }
        return new Time(h, m, s);
    }
    int toSeconds() {
        return (this.hour * 60 * 60 + this.min * 60 + this.sec);
    }
    double toHours() {
        return (((double)(this.hour)) + (((double)(this.min)) / 60) +
(((double)(this.sec)) / (60 * 60)));
    }
}
public class Q1 {
    public static void main(String[] args) {
        Time t1 = new Time(5, 20, 15);
        Time t2 = new Time(3, 45, 30);
        Scanner sc = new Scanner(System.in);
        System.out.println("Operations With Command :");
        System.out.println("1. Display Time \n2. Set Time \n3. Add Time \n4.
Subtract Time \n5. Convert to Seconds \n6. Convert to Hours \n7. Exit");
        while (true) {
            System.out.println("");
            System.out.print("Enter the number corresponding to operation to be
performed : ");
            int ops = sc.nextInt();
            if(ops == 1){
```

```
    System.out.println("Timings Are : ");
    t1.displayTime();
    t2.displayTime();
}
else if(ops == 2){
    System.out.println("Enter the hour, minutes and seconds : ");
    int hr = sc.nextInt();
    int m = sc.nextInt();
    int s = sc.nextInt();
    t1.setTime(hr, m, s);
    System.out.println("Updated Time : ");
    t1.displayTime();
}
else if(ops == 3){
    Time t3 = t1.addTime(t1, t2);
    System.out.println("New Time : ");
    t3.displayTime();
}
else if(ops == 4){
    Time t4 = t1.subtractTime(t1, t2);
    System.out.println("New Time : ");
    t4.displayTime();
}
else if(ops == 5){
    System.out.println("In Seconds : " + t1.toSeconds() + " Seconds");
}
```

```
         else if(ops == 6){
            System.out.println("In Hours : " + t1.toHours() + " Hours");
         }
         else if(ops == 7){
            break;
         }
      }
   }
}
```

---

b) Define a class Item having the private data members; itemcode, itemname, category, price, discount, and quantity. In addition, define a constant MAXQUANTITY to set a uppper limit of the stock of the item. The class must have folllowing methods:

→ A default constructor

→ A parameterized constructor

→ A displayitem() method to display the detail of an item. → An updateDetail(price,discount) to update the price and discount rate of an item.

→ An addItems(n) method to add n number of items to quantity field provided total quantity does not go beyond MAXQUANTITY.

→ A sellItem(n) method to sell n number of items to customer(s) provided number of item requested is available.

→ A compareItem(Item) method to compare and display the fields of two item in an appropriate format

<u>CODE</u> :-

```
import java.util.Scanner;
class Item {
    private int itemcode, quantity;
    private double discount, price;
    private String itemname, category;
    final int maxQuantity = 1000;

    Item() {
        itemcode = 0;
        itemname = "Item_0";
        price = 100;
        discount = 0;
        quantity = 10;
        category = "Food";
    }

    Item(int itemcode, String itemname, double price, double discount, int
quantity, String category) {
        this.itemcode = itemcode;
        this.itemname = itemname;
        this.price = price;
        this.discount = discount;
        this.quantity = quantity;
        this.category = category;
    }

    void displayDetail() {
        System.out.println("Item Details : ");
        System.out.println("Itemcode : " + itemcode);
        System.out.println("Itemname : " + itemname);
        System.out.println("Price : $" + price);
        System.out.println("Discount : " + discount + "%");
        System.out.println("Quantity : " + quantity);
        System.out.println("Category : " + category);
        System.out.println("");
    }

    void updateDetail(double price, double discount) {
        this.price = price;
        this.discount = discount;
    }

    void addItems(int n) {
        if ((quantity + n) > maxQuantity) {
            System.out.println("Max Quantity of items exceeded!");
            System.out.println("");
        } else {
            quantity += n;
        }
    }

    void sellItem(int n) {
        if ((quantity - n) < 0) {
            System.out.println("Demand is more than the available items!");
            System.out.println("");
        } else {
            quantity -= n;
        }
    }

    void compareItem(Item other) {
        System.out.println("Comparing Items:");
        System.out.println("");
        System.out.println("Item 1 - " + this.itemname + " | Item 2 - " +
other.itemname);
        System.out.println("Price: $" + this.price + " vs $" + other.price);
        System.out.println("Discount: " + this.discount + "% vs " + other.discount +
"%");
        System.out.println("Quantity: " + this.quantity + " vs " + other.quantity);
        System.out.println("Category: " + this.category + " vs " + other.category);
        System.out.println("");
    }
}
public class Q2 {
    public static void main(String[] args) {
        Item i1 = new Item();
        Item i2 = new Item(101, "Laptop", 5000, 10, 20, "Electronics");
        Scanner sc = new Scanner(System.in);
        System.out.println("Operations With Command :");
        System.out.println("1. Display Item \n2. Update Details \n3. Add Item \n4.
Sell Item \n5. Compare Item \n6. Exit");
        while (true) {
            System.out.print("Enter the number corresponding to operation to be 
performed : ");
            int ops = sc.nextInt();
            if(ops == 1){
                System.out.println("Items Are : \n");
                i1.displayDetail();
                i2.displayDetail();
            }
            else if(ops == 2){
                System.out.println("Enter the price and discount");
                int price = sc.nextInt();
                int discount = sc.nextInt();
                i1.updateDetail(price , discount);
                System.out.println("Updated Item : ");
                i1.displayDetail();
            }
            else if(ops == 3){
                System.out.println("Enter the number of items : ");
                int n = sc.nextInt();
                i2.addItems(n);
                i2.displayDetail();
            }
            else if(ops == 4){
                System.out.println("Enter the number of items : ");
                int n = sc.nextInt();
                i2.sellItem(n);
                i2.displayDetail();
            }
            else if(ops == 5){
               i1.compareItem(i2);
            }
            else if(ops == 6){
               break;
            }
        }
    }
}
```

---

c) Define a class Battery having data field: level, which indicate its energy level. When an object is instantiated it has 100 units battery level. The class must implement the following methods:

→ A default constructor

→ showLevel() method to show the current battery level. → sendMsg() method is used to consume 2 units of battery on each call

→ recvMsg() method is used to consume 1 unit of battery on each call

→ compute() method consumes 1.5 unit of battery on each call

→ recharge(minutes) method update the battery level , 1 unit per 2 minutes.

<u>CODE</u> :-

```
import java.util.Scanner;
class Battery{
   double level;
   Battery(){
      level = 100;
```

```
}
void showLevel(){
    System.out.println("Current Battery Level : " + level);
    System.out.println("");
}
void sendMsg(){
    if(level >= 2){
        level -= 2;
        System.out.println("Message Sent!");
    }
    else{
        System.out.println("Insufficient Battry to perfrom the Operation!");
    }
}
void recvMsg(){
    if(level >= 1){
        level -= 1;
        System.out.println("Message Received!");
    }
    else{
        System.out.println("Insufficient Battry to perfrom the Operation!");
    }
}
void compute(){
    if(level >= 1.5){
        level -= 1.5;
```

```
            System.out.println("Computation Done!");
        }
        else{
            System.out.println("Insufficient Battry to perfrom the Operation!");
        }
    }
    void recharge(int minutes){
        double rech = minutes / 2;
        if(rech > (100 - level)){
            rech = (double)(100 - level);
        }
        level += rech;
        System.out.println("Battery Recharged!");
    }
}
public class Q3 {
    public static void main(String[] args) {
        Battery b1 = new Battery();
        Scanner sc = new Scanner(System.in);
        System.out.println("Operations With Command :");
        System.out.println("1. Show Level \n2. Send Message \n3. Receive
Message \n4. Compute \n5. Recharge \n6. Exit");
        while (true) {
            System.out.println("");
            System.out.print("Enter the number corresponding to operation to be
performed : ");
            int ops = sc.nextInt();
        if(ops == 1){
            b1.showLevel();
        }
        else if(ops == 2){
            b1.sendMsg();
            b1.showLevel();
        }
        else if(ops == 3){
            b1.recvMsg();
            b1.showLevel();
        }
        else if(ops == 4){
            b1.compute();
            b1.showLevel();
        }
        else if(ops == 5){
            System.out.println("Enter the charging time(In Minutes) : ");
            int n = sc.nextInt();
            b1.recharge(n);
            b1.showLevel();
        }
        else if(ops == 6){
            break;
        }
    }
}
```

```
}
```

---

d) Implement a class MyByte to add a data member with byte datatypes. Implement and execute all the following member functions provided below:

→ A default constructor (set to 0)

→ A parameterized constructor

→ andOP() method to perform bitwise AND between two Byte objects.

→ orOP() method to perform bitwise OR between to Byte objects. → xorOP() method to perform bitwise XOR betwween to MyByte objects.

→ mask() method to mak a set of bits of a given MyByte object.

→ complement() method complement all the bits of a MyBite object.

CODE :-

```
import java.util.Scanner;
class MyByte {
   byte data;
   MyByte(){
      data = 0;
   }
   MyByte(byte data){
      this.data = data;
   }
   void andOp(MyByte b1){
      byte ans = ((byte)(b1.data & this.data));
      System.out.println("Result : " + ans);
   }
```

```
    void orOp(MyByte b1){
        byte ans = ((byte)(b1.data | this.data));
        System.out.println("Result : " + ans);
    }

    void xorOp(MyByte b1){
        byte ans = ((byte)(b1.data ^ this.data));
        System.out.println("Result : " + ans);
    }
    void mask(byte b){
        data = ((byte)(b & this.data));
        System.out.println("Result : " + data);
    }

    void complement(){
        data = ((byte)(~(this.data)));
        System.out.println("Result : " + data);
    }
}
public class Q4 {
    public static void main(String[] args) {
        MyByte b1 = new MyByte();
        MyByte b2 = new MyByte();
        Scanner sc = new Scanner(System.in);
        System.out.println("Operations With Command :");
    System.out.println("1. Set Object1 \n2. Set Object2 \n3. And Operation
\n4. Or Operation \n5. Xor Operation \n6. Masking \n7. Complement
\n8.Exit");
    while (true) {
      System.out.println("");
      System.out.print("Enter the number corresponding to operation to be
performed : ");
      int ops = sc.nextInt();
      if(ops == 1){
        System.out.print("Enter the data : ");
        byte n = sc.nextByte();
        b1 = new MyByte(n);
      }
      else if(ops == 2){
        System.out.print("Enter the data : ");
        byte n = sc.nextByte();
        b2 = new MyByte(n);
      }
      else if(ops == 3){
        b1.andOp(b2);
      }
      else if(ops == 4){
        b1.orOp(b2);
      }
      else if(ops == 5){
        b1.xorOp(b2);
      }
```

```
            else if(ops == 6){
                System.out.print("Enter a number for masking : ");
                byte n = sc.nextByte();
                b1.mask(n);
            }
            else if(ops == 7){
                b2.complement();
            }
            else if(ops == 8){
                break;
            }
        }
    }
}
```
