

Object-Oriented Design Lab Report (Java 1)

Bisakh Mondal 001810501079

Format: Question | Approach(if notable) | Code | Output

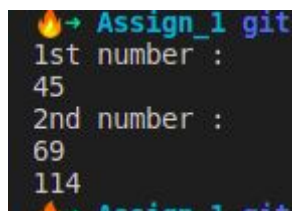
Java Assignment

(Q1) Write a program to accept two short integers from user and display the sum.

```
import java.util.Scanner;

class Sum{

public static void main(String []argc){
    short a,b;
    Scanner sc=new Scanner(System.in);
    System.out.println("1st number :");
    a=sc.nextShort();
    System.out.println("2nd number :");
    b=sc.nextShort();
    int sum=a+b;
    System.out.println(sum);
    sc.close();
}
}
```



```
→ Assign_1 git
1st number :
45
2nd number :
69
114
→ Assign_1 git
```

(Q2) Write a program that accepts number of command line parameters and displays the parameters and count of such parameters.

```
class Cline{
public static void main(String []args){
    for(String s:args){
```

```

    System.out.println(s);
}
    System.out.println("NUMBER OF ARGS: "+args.length);
}
}

```

```

4
🔥 → Assign_1 git:(4TH
Bisakh
Mondal
NUMBER OF ARGS: 2
4
🔥 → Assign_1 git:(4TH

```

(Q3) Write a program that accepts height in cm as int and displays the height in feet and inches. Assume, 1 inch equals to 2.54 cm and 1 foot equals to 30.5 cm.

```

import java.util.Scanner;
class Height{
public static void main(String [] args){

    final float c2f=30.5f;
    final float c2i=2.54f;

    System.out.println("Enter Height in cm");
    int height;
    Scanner sc=new Scanner(System.in);
    height=sc.nextInt();
    int feet=(int)(height/c2f);
    int inch=(int)((height-(feet*c2f))/c2i);
    System.out.println("foot: "+feet+" inch: "+inch);
    sc.close();
}
}

```

```

🔥 → Assign_1 git:(4TH
Height
Enter Height in cm
450
foot: 14 inch: 9
🔥 → Assign_1 git:(4TH

```

(Q4) Write a program that accepts radius of a circle and displays area of the circle. Declare a constant pi equals to 3.14.

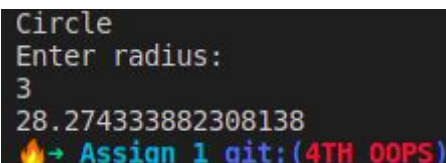
```
import java.util.Scanner;

class Circle{
public static void main(String [] args){
    System.out.println("Enter radius: ");
    Scanner sc=new Scanner(System.in);
    int radius=sc.nextInt();

    final double pi=Math.PI;

    double area=pi*Math.pow(radius,2);

    System.out.println(area);
}
}
```



```
Circle
Enter radius:
3
28.274333882308138
🔥 -> Assign 1 git: (4TH OOPS)
```

(Q5) Write a program that accepts a String and assigns it to another. Check the outcome of comparison with == and equals() method. Take two Strings and put same input for them. Repeat the equality checking. Observe the outcome.

In java '==' checks for equality in reference or same object but 'equals()' check for content equality.

```
import java.util.Scanner;

class Check{
public static void main(String args[]){
    System.out.println("Enter first String: ");
    Scanner sc=new Scanner(System.in);
    String ss=sc.next(), ss1;
    ss1=ss;

    EqualsCheck(ss, ss1);
}
```

```

System.out.println("Enter Strings: ");
String s=sc.next();
String s1=sc.next();

EqualsCheck(s, s1);
}

static void EqualsCheck(String s, String s1){
    if(s==s1){
        System.out.println("== Equal");
    }
    else{
        System.out.println("== !Equal");
    }
    if(s.equals(s1)){
        System.out.println(" Equal");
    }
    else{
        System.out.println(" !Equal");
    }
    System.out.println("-----");
}
}

```

```

Enter first String:
bbbbbbbbb
== Equal
Equal
-----
Enter Strings:
Bisakh
Bisakh
== !Equal
Equal
-----
🔥 → Assign 1 git: (4TH

```

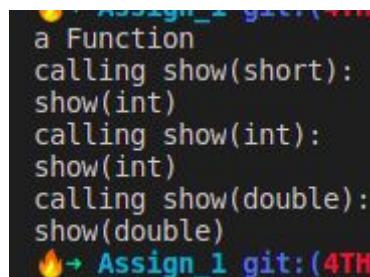
(Q6) Write a program where class contains **void show(int)** to display the argument passed. Call the function once with **short** as actual parameter and again **double** as actual parameter. Add another function as **void show(double)** . Repeat the calls. Observe the outcomes in each case.

Java implicitly supports only Widening conversion from one to another data type, Narrowing conversion is not allowed, have to convert it explicitly.

```
class Function{
public static void main(String args[]){
    int a=5;
    short s=1;
    double d=5.6;

    System.out.println("calling show(short): ");
    show(s);
    System.out.println("calling show(int): ");
    show(a);
    System.out.println("calling show(double): ");
    show(d);

}
public static void show(double a){
    System.out.println("show(double)");
}
public static void show(int a){
    System.out.println("show(int)");
}
}
```



```
a Function
calling show(short):
show(int)
calling show(int):
show(int)
calling show(double):
show(double)
```

(Q7) Design and implement Student class with roll, name and score as attributes. It will have methods to set attributes (attribute values passed as arguments), display the attributes, copy (that copies the content of invoking object to another object passed as argument). Verify that methods are working properly.

The statement **Student s1=s** is an example of a shallow copy, the same object is referenced through another variable name.

```

class StudentClass{

    public static void main(String[] args) {
        StudentClass ss=new StudentClass();
        ss.start();
        Student s=new Student();
        s.set("Bisakh mondal","079",100);
        s.display();
    }
    public void start(){
        Student s=new Student();
        s.set("Bisakh","079",100);
        s.display();
        Student s1=s;//shallow copy
        s1.score=96;
        s.display();

    }
}

class Student{
    /*private*/ int score;
    private String roll,name;
    public void set(String name,String roll,int score){
        this.setRoll(roll);
        this.setName(name);
        this.score=score;
    }
    public void display(){
        System.out.println("Name: "+getName()+" Roll:
"+getRoll()+" :score: "+score);
    }
    public void copy(Student c){
        score=c.score;
        setRoll(c.getRoll());
        setName(c.getName());
    }
    public int getScore() {
        return score;
    }
    public String getName() {
        return name;
    }
}

```

```

    }
    public void setName(String name) {
        this.name = name;
    }
    public String getRoll() {
        return roll;
    }
    public void setRoll(String roll) {
        this.roll = roll;
    }
    public void setScore(int score) {
        this.score = score;
    }
}

```

```

java StudentClass
Name: Bisakh Roll: 079 :score: 100
Name: Bisakh Roll: 079 :score: 96
Name: Bisakh mondal Roll: 079 :score: 100
🔥 → Assign 1 nit: (4TH OOPS) x □

```

(Q8) Add constructors in the Student class of earlier problem so that objects can be created with i) roll only, ii) roll and name only, iii) roll, name and score, iv) no value. Also include a copy constructor. Check whether constructors are working or not. Verify, copy constructor results into deep copy or not.

```

class StudentModified{

    public static void main(String[] args) {

        Student s=new Student();
        s.setName("Bisakh");
        s.setScore(100);
        s.setRoll("079");
        s.display();
        Student s1=s;//shallow copy
        s1.score=96;
        s1.display();
        Student s2=new Student(s);//Deep copy
    }
}

```

```

        s2.score=9;
        s.display();
    }
}
class Student{
    int score;
    private String roll,name;
    Student(String roll){
        this.roll=roll;
    }
    Student(String roll,String name){
        this.roll=roll;
        this.name=name;
    }
    Student(String roll,String name,int score){
        this.roll=roll;
        this.name=name; this.score=score;
    }
    Student(){};

    Student(Student c){
        name=c.name;
        roll=c.roll;
        score=c.score;
    }
    public void display(){
        System.out.println("Name: "+name+" Roll: "+roll+" :score:
"+score);
    }
    public void copy(Student c){
        score=c.score;
        roll=c.roll;
        name=c.name;
    }
    public int getScore() {
        return score;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }

```



```

    }
    public String getRoll() {
        return roll;
    }
    public void setRoll(String roll) {
        this.roll = roll;
    }
    public void setScore(int score) {
        this.score = score;
    }
}

```

```

&& java StudentModified
Name: Bisakh Roll: 079 :score: 100
Name: Bisakh Roll: 079 :score: 96
Name: Bisakh Roll: 079 :score: 96

```

(Q9) Design a BankAcct class with account number, balance and interest rate as attribute. Interest rate is same for all account. Support must be there to initialize, change and display the interest rate. Also supports are to be there to return balance and calculate interest.

```

class BankAcct {
    private String acc_num;
    private double balance;
    private static double interestRate = 4.0;

    BankAcct(final String num, final double balance) {
        setAcc_num(num);
        this.balance = balance;
    }

    public String getAcc_num() {
        return acc_num;
    }

    public void setAcc_num(String acc_num) {

```

```

        this.acc_num = acc_num;
    }

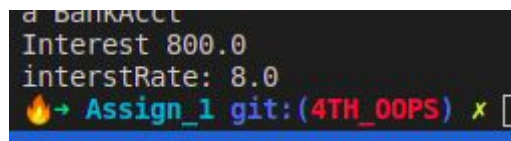
    static void setInterest(final double rate) {
        interestRate = rate;
    }

    static double getInterest() {
        return interestRate;
    }

    public double interest() {
        return balance * interestRate / 100;
    }

    public static void main(final String[] args) {
        final BankAcct acc1 = new BankAcct("00156", 10000);
        BankAcct.setInterest(8.0);
        final double interest = acc1.interest();
        System.out.println("Interest "+interest);
        System.out.println("interstRate:
"+BankAcct.getInterest());
    }
}

```



```

a BankAcct
Interest 800.0
interstRate: 8.0
🔥 → Assign_1 git:(4TH_OOPS) x

```

(Q10) Design a Metric class that supports Kilometre to Mile conversion with distance in Kilometre as argument and Mile to Kilometre conversion with distance in mile as argument. Assume, one Mile equals 1.5 Kilometre.

```

class Metric{
    static final double conv_rate=1.5;
    public static double mile2km(double mile){
        return mile*conv_rate;
    }
    public static double km2mile(double km){
        return km/conv_rate;
    }
}

```

```

    }
    public static void main(String[] args) {
        System.out.println("12 mile: "+mile2km(12));
        System.out.println("18 km: "+km2mile(18));
    }
}

```

```

metric
12 mile: 18.0
18 km: 12.0
🔥 → Assign_1 g

```

(Q11) Each Instructor has name and phone number. One can view instructor information and set the information. Textbook has a title, author name and publisher. One can set the data for a textbook and view the same. Each course has a course name, instructor and text book. One can set the course data and view the same. Design and implement the classes

```

class Instructor {
    private String name, phone_number;

    public String getName() {
        return name;
    }

    public String getPhone_number() {
        return phone_number;
    }

    public void setPhone_number(String phone_number) {
        this.phone_number = phone_number;
    }

    public void setName(final String name) {
        this.name = name;
    }

    public void print(){
        System.out.println("Name: "+name+" phone_number: "+phone_number);
    }
}

```

```

}

class Textbook {
    private String author, title, publisher;

    public String getAuthor() {
        return author;
    }

    public String getPublisher() {
        return publisher;
    }

    public void setPublisher(final String publisher) {
        this.publisher = publisher;
    }

    public String getTitle() {
        return title;
    }

    public void setTitle(final String title) {
        this.title = title;
    }

    public void setAuthor(final String author) {
        this.author = author;
    }

    public void print(){
        System.out.println("title: "+title+" author: "+author+"
Publisher: "+publisher);
    }
}

class Course {
    private String course_name;
    private Textbook book;
    private Instructor i;
    Course(String name){
        course_name=name;
    }
    public String getCourse_name() {

```

```

        return course_name;
    }

    public Textbook getBook() {
        return book;
    }

    public void setBook(Textbook book) {
        this.book = book;
    }

    public Instructor getI() {
        return i;
    }

    public void setI(Instructor i) {
        this.i = i;
    }

    public void setCourse_name(final String course_name) {
        this.course_name = course_name;
    }

    public void print(){
        System.out.println("Course_name: "+course_name);
        book.print();
        i.print();
    }

    public static void main(String[] args) {
        Textbook t=new Textbook();
        t.setAuthor("Doyle");
        t.setPublisher("abc");
        t.setTitle("sherlock");
        Course c=new Course("Literature");
        c.setBook(t);
        Instructor i=new Instructor();
        i.setName("piSigma");
        i.setPhone_number("987654321");
        c.i=i;
        c.print();
    }
}

```

```
Course
Course_name: Literature
title: sherlock author: Doyle Publisher: abc
Name: piSigma phone_number: 987654321
🔥➡Assign 1 pit:(4TH DOPS) ✖📦
```