SUNWAY

INT'L BUSINESS SCHOOL

Programme Name: BCS HONS

Course Code: CSC 2515

Course Name: Object Oriented Programming

Classes and Object Labsheet

Date of Submission: 2/10/2020

Submitted By:

Student Name**: Dipesh Tha Shrestha**

IUKL ID: **041902900028**

Semester**: Third Semester**

Intake**: September 2019**

Submitted To:

Faculty Name**: Som Prasad Shrestha**

Department**: LMS**

1. Write a program that implements Interest class which stores three variables principal, rate and time and calculates simple interest. Create constructor(s) to initialize the variables. Write main method to test the class.

Answer:

```
 import java.util.Scanner;
public class Interest{
   double principal;float rate;float time;
     Interest(double principal,float rate,float time)
   {
      this.principal =principal;
      this.rate =rate;
      this.time =time;
   }
 public  double getPrincipal(){
return principal;
}
public  float getRate(){
return rate;
}
public  float getTime(){
return time;
}
public void setPrincipal(double principal){
```

```
this.principal =principal;
}
public void setRate (float rate){
this.rate =rate;
}
public void setTime (float time){
this.time =time;
}
     void calculateInterest(){
     double SimpleInt = this.principal * this.rate *this.time / 100;
     System.out.println("The Interest obj is "+ SimpleInt);
    }
    public static void main(String[] args)
    {
      Scanner len=new Scanner(System.in);
      System.out.println("Enter the following requirement to calculate the simple interest");
      System.out.println("Enter the principal:");
      double p=len.nextDouble();
      System.out.println("Enter the rate(in %):");
      float r =len.nextFloat();
      System.out.println("Enter the year(in year)");
      float t=len.nextFloat();

      Interest obj=new Interest(p,r,t);

      obj.calculateInterest();


   }

}
```

2. Design a class to represent a bank account. Include the following members:
   - Fields/Data members
     - Name of the depositor
     - Account number
     - Type of account
     - Balance amount in the account
   - Methods
     - Constructor(s)
     - To assign initial values
     - To deposit an amount
     - To withdraw an amount after checking balance
     - To display the name and balance

Answer :

```
 import java.util.Scanner;

public class Bank{

String Name, AccountNumber, TypeofAccount;

double Balance_amount;

Bank(String Name,String AccountNumber,String TypeofAccount,double
Balance_amount)
```

```
{
this.Name=Name;
this.AccountNumber=AccountNumber;
this.TypeofAccount=TypeofAccount;
this.Balance_amount=Balance_amount;
}
     double Deposit(Scanner len){
    System.out.println("Enter the deposit amount");
     double deposit=len.nextDouble();
       this.Balance_amount = this.Balance_amount + deposit;
       System.out.println("Your deposit amount is "+deposit);
       System.out.println("Total balance in your account is "+this.Balance_amount);
       return this.Balance_amount;
    }
    double Withdraw(Scanner len){
                System.out.println("Enter the withdraw amount");
               double withdraw=len.nextDouble();
       if(withdraw < this.Balance_amount)
       {
          this.Balance_amount = this.Balance_amount - withdraw;
          System.out.println("Your withdraw amount is "+withdraw);
       System.out.println("Total balance in your account is "+this.Balance_amount);
```

```
        }
        else
          System.out.println("Sorry,You Don't Have Enough balance");
        return this.Balance_amount;
    }
    void Display(){

System.out.println("********************YourStatement************************");
        System.out.println("Account Holder's Name is "+this.Name);
        System.out.println("Total balance in your account is "+this.Balance_amount);
        System.out.println("Thank you for visting us");
    }
    public static void main(String[] args)
    {
       String Name, AccountNumber, TypeofAccount;
        double Balance_amount;
        Scanner len =new Scanner(System.in);
       System.out.println("Account Holder's Name(in string): ");
       Name = len.next();
        System.out.println("Account Number(in string): ");
        AccountNumber=len.next();
       System.out.println("Account Type(in string): ");

       TypeofAccount =len.next();

        System.out.println("Initial Balance(in double): ");

       Balance_amount =len.nextDouble();

      Bank acc1 =new Bank(Name, AccountNumber, TypeofAccount,
Balance_amount);

      acc1.Deposit(len);

      System.out.println("=============================================");

      acc1.Withdraw(len);

      System.out.println("=============================================");

      acc1.Display();

    }

}
```

3. Define a class Student as described below:
Instance variables:
Name , age, marks in three subjects (m1, m2, m3), maximum and average.
Methods:
i. A parameterized constructor to initialize the instance variables.
ii. To accept the details of a student.
iii. To compute the average and minimum out of the three marks.

iv. To display the name, age, marks in the three subjects, minimum and average.

v. Write a main method to create an object of the class and call the above methods.

Answer:

```
import java.util.Scanner;
public class Student{
        String Name;
        int Age;
        String subject[];
        double marks[];
        double maximum;
        double average;
Student(String Name,int Age,int len)
{
        this.Name=Name;
        this.Age=Age;
        this.subject=new String[len];
        this.marks=new double[len];
}
void setMarks(Scanner sc)
{
for (int i = 0; i < subject.length; i++) {
System.out.println("Subject [" + (i + 1) + "](in string):");
subject[i] = sc.next();
System.out.println("Marks in Subject [" + (i + 1) + "](in double):");
marks[i] = sc.nextDouble();
}
}
double getTotal() {
double sum = 0;
for (int i = 0; i < subject.length; i++) {
sum = sum + marks[i];
}
return sum;
}
double getAverage(double sum) {
double average = sum / subject.length;
return average;
}
double getMinimum()
{
double minimum = marks[0];
for (int i = 1; i < marks.length; i++) {
if (marks[i] < minimum) {
minimum = marks[i];
}
}
return minimum;
}
String getMinimumSubject() {
String subjects = null;
double minimum = marks[0];
for (int i = 1; i < marks.length; i++) {
if (marks[i] < minimum) {
subjects = subject[i];
}
}
return subjects;
}
public void display() {
System.out.println("*********************Students
Records*********************");
System.out.println("Name of the student is " + this.Name);
System.out.println("Age of the student is " + this.Age);
System.out.println("=============================================
==============");
System.out.println("Subjects\t\tMarks");
for (int i = 0; i < this.marks.length; i++) {
System.out.println(this.subject[i] + "\t\t\t" + this.marks[i]);
}
double total = this.getTotal();
double average = this.getAverage(total);
System.out.println("=============================================
=============");
System.out.println("Total marks obtained by Student: "+ total);
System.out.println("Average marks of the Student: " + average);
System.out.println("Minimum marks obtained by the Student in "
+getMinimumSubject()+ " with marks " + getMinimum());
}
public static void main(String[] args) {
Scanner len = new Scanner(System.in);
String Name;
int Age;
int sub;
System.out.println("Enter the following instances: ");
System.out.println("Name(String): ");
Name = len.next();
System.out.println("Age(int): ");
Age = len.nextInt();
System.out.println("Number of subjects(int): ");
sub = len.nextInt();
Student stud = new Student(Name, Age, sub);
stud.setMarks(len);
stud.display();
}
}
```

4. Define a constructor for ShowDate class that initializes the ShowDate objects with given initial values. In case the initial values are not provided, it should initialize the object with default 0 values.

Answer:

```
import java.util.Scanner;
public class Showdate {
int date;
Showdate(int date) {
this.date = date;
}
Showdate() {
this.date = 0;
}
void display() {
System.out.println("Display Date: " + this.date);
}
public static void main(String[] args) {
Scanner len = new Scanner(System.in);
System.out.println("Date(int): ");
int date = len.nextInt();
System.out.println("============================================");
Showdate obj1 = new Showdate(date);
obj1.display();
System.out.println("============================================");
Showdate obj2 = new Showdate();
obj2.display();
}
}
```

```
}
```

5. Write a class Circle with a parameterized constructor. If no parameters are passed then its default constructor should be invoke parameterized constructor with default values. A circle is defined using radius and circumference.

Answer:

```
import java.util.Scanner;
public class Circle {
static double pi = 3.14;
double radius;
Circle(){
this.radius = 2;
}
Circle(double radius){
this.radius = radius;
}
public double calcCircumference(){
double circumference = 2*(pi)*this.radius;
return circumference;
}
public static void main(String[] args) {
Scanner sc = new Scanner(System.in);
System.out.println("Enter the radius of a circle: ");
double r = sc.nextDouble();
Circle obj1 = new Circle(r);
double circumference = obj1.calcCircumference();
System.out.println("The circumference of a circle with radius " +r+ " is: "
+circumference);
Circle obj2 = new Circle();
circumference = obj2.calcCircumference();
System.out.println("The circumference of a circle with default radius is: "
+circumference);
}
}
```

6. Write a program to implement a Book class that stores the details of a book namely, bookcode, name of the book, name of the author(s) and price. The class has methods to display any of the details individually.

Answer:

```
import java.util.Scanner;
public class Book{
    String Bookcode;
    String Bookname;
    String Authorname;
    double price;
Book(String Bookcode,String Bookname,String Authorname,double price)
{
 this.Bookcode=Bookcode;
this.Bookname=Bookname;
this.Authorname=Authorname;
this.price=price;
}
void displaybookcode()
{
    System.out.println("Book code is "+this.Bookcode);
}
void displaybookname()
{
    System.out.println("Book name is "+this.Bookname);
}
void displayauthorname()
{
    System.out.println("Book authorname is "+this.Authorname);
}
void displayprice()
{
      System.out.println("Book price is "+this.price);
}
public static void main(String[] args)
{
  Scanner len=new Scanner(System.in);
 System.out.print("Enter the Bookcode(in string)");
 String Bookcode=len.nextLine();
 System.out.print("Enter the Bookname(in string)");
 String Bookname=len.nextLine();
 System.out.print("Enter the Authorname(in string)");
 String Authorname=len.nextLine();
 System.out.print("Enter the price(in double)");
 double price=len.nextDouble();
 System.out.println("======================================");
  Book obj=new Book(Bookcode,Bookname,Authorname,price);
  obj.displaybookcode();
  System.out.println("======================================");
  obj.displaybookname();
  System.out.println("======================================");
  obj.displayauthorname();
  System.out.println("======================================");
  obj.displayprice();
}
}
```

7. Implement a class called Dimension based on the following information:
   Constructors
   Dimension(double length, double width, double height)
   Dimension(double side)
   Methods

   double volume() // length*width*height

```
double area() // 2*(length*width+width*height+height*length)
```

Make all the instance variables private so that they can be accessed only by the methods defined within the class. Make the methods public. Test your program.

Answer:

```
import java.util.Scanner;
public class Dimension {
private double length;
private double width;
private double height;
private double side;
Dimension(double length, double width, double height) {
this.length = length;
this.width = width;
this.height = height;
}
Dimension(double side) {
this.length = side;
this.width = side;
this.height = side;
}
public double calcVolume() {
double volume = this.length * this.width * this.height;
return volume;
}
public double calcArea() {
double area = 2 * (this.length * this.width + this.width* this.height +
this.height * this.length);
return area;
}
public static void main(String[] args) {
Scanner len = new Scanner(System.in);
System.out.println("Enter the folloing instances: ");
System.out.print("Length: ");
double l = len.nextDouble();
System.out.print("Width: ");
double w = len.nextDouble();
System.out.print("Height: ");
double h = len.nextDouble();
System.out.print("Side: ");
double s = len.nextDouble();
Dimension cuboid1 = new Dimension(l, w, h);
System.out.println("The volume of cuboid having length, width, height of "
+l+", " +w+", " +h+" respectively is " +cuboid1.calcVolume());
System.out.println("The area of cuboid having length, width, height of "
+l+", " +w+", " +h+" respectively is " +cuboid1.calcArea());
Dimension cube1 = new Dimension(s);
System.out.println("The volume of cube having side of " +s+" is "
+cube1.calcVolume());
System.out.println("The area of cube having side of " +s+" is "
+cube1.calcArea());
}
}
```

8. Modify the implementation of area() given in the previous question using private methods, faceArea(), topArea() and sideArea(). [Often private methods are helping methods that public methods use, but are not to be used outside the class.] Test your program.

   Answer:

```
import java.util.Scanner;
public class Dimension {
private double length;
private double width;
private double height;
private double side;
Dimension(double length, double width, double height) {
this.length = length;
this.width = width;
this.height = height;
}
Dimension(double side) {
this.length = side;
this.width = side;
this.height = side;
}
public double calcVolume() {
double volume = this.length * this.width * this.height;
return volume;
}
private double faceArea() {
double faceArea = this.width * this.height;
return faceArea;
}
private double topArea() {
double topArea = this.length * this.width;
return topArea;
}
private double sideArea() {
double sideArea = this.length * this.height;
return sideArea;
}
public double calcArea() {
double area = 2 * (faceArea() + topArea() + sideArea());
return area;
}
```

```
public static void main(String[] args) {
Scanner sc = new Scanner(System.in);
System.out.println("Enter the folloing instances: ");
System.out.print("Length: ");
double l = sc.nextDouble();
System.out.print("Width: ");
double w = sc.nextDouble();
System.out.print("Height: ");
double h = sc.nextDouble();
System.out.print("Side: ");
double s = sc.nextDouble();
Dimension cuboid1 = new Dimension(l, w, h);
System.out.println("The volume of cuboid having length, width, height of "
+ l + ", " + w + ", " + h+ " respectively is " + cuboid1.calcVolume());
System.out.println("The area of cuboid having length, width, height of " + l
+ ", " + w + ", " + h+ " respectively is " + cuboid1.calcArea());
Dimension cube1 = new Dimension(s);
System.out.println("The volume of cube having side of " +s + " is " +
cube1.calcVolume());
System.out.println("The area of cube having side of " + s+ " is " +
cube1.calcArea());
}
}
```

9. Add a new constructor to the Dimension class created in question 1 as Dimension(Dimension dim) This constructor creates a new Dimension object with identical dimensions as the old Dimension object. The old object is not changed.

Answer:

```
import java.util.Scanner;
public class Dimension {
private double length;
private double width;
private double height;
private double side;
Dimension(double length, double width, double height) {
this.length = length;
this.width = width;
this.height = height;
}
Dimension(double side) {
this.length = side;
this.width = side;
this.height = side;
}
Dimension(Dimension dim) {
this.length = dim.length;
this.width = dim.width;
this.height = dim.height;
}
public double calcVolume() {
double volume = this.length * this.width * this.height;
return volume;
}
public double calcArea() {
double area = 2 * (this.length * this.width + this.width
* this.height + this.height * this.length);
return area;
}
public static void main(String[] args) {
Scanner sc = new Scanner(System.in);
System.out.println("Enter the folloing instances: ");
System.out.print("Length: ");
double l = sc.nextDouble();
System.out.print("Width: ");
double w = sc.nextDouble();
System.out.print("Height: ");
double h = sc.nextDouble();
System.out.print("Side: ");
double s = sc.nextDouble();
Dimension cuboid1 = new Dimension(l, w, h);
System.out.println("The volume of cuboid1 having length,width, height of " + l
+ ", " + w + ", " + h+ " respectively is " + cuboid1.calcVolume());
System.out.println("The area of cuboid1 having length, width, height of " + l +
", " + w + ", " + h+ " respectively is " + cuboid1.calcArea());
Dimension cube1 = new Dimension(s);
System.out.println("The volume of cube1 having side of "+ s + " is " +
cube1.calcVolume());
System.out.println("The area of cube1 having side of " +s + " is " +
cube1.calcArea());
System.out.println("=====================================================
===============================================");
Dimension cuboid2 = new Dimension(cuboid1);
System.out.println("The volume of cuboid2 having length,width, height of " + l
+ ", " + w + ", " + h+ " respectively is " + cuboid2.calcVolume());
System.out.println("The area of cuboid2 having length, width, height of " + l +
", " + w + ", " + h+ " respectively is " + cuboid2.calcArea());
Dimension cube2 = new Dimension(cube1);
System.out.println("The volume of cube2 having side of "+ s + " is " +
cube2.calcVolume());
System.out.println("The area of cube2 having side of " +s + " is " +
cube2.calcArea());
}
}
```

10.Design a class to overload a function num_calc() as follows:

i. void num_calc (int num) with one integer argument, computes the square of integer argument.

ii. void num_calc (int a, int b) with two integer arguments. It computes the product of integer arguments.

iii. void num_calc (String s1, String s2) with two string arguments, which concatenates and prints the combined strings.

Answer:

```
import java.util.Scanner;
public class OverLoad {
int num;
int a, b;
String s1, s2;
public void num_calc(int num) {
int sqr = num * num;
System.out.println("The square of " +num+ " is " +sqr);
}
public void num_calc(int a, int b) {
int product = a * b;
System.out.println("The product of " +a+ " & " +b+ " is "+product);
}
public void num_calc(String s1, String s2) {
String concatString = s1.concat(s2);
System.out.println("The concatenation of " +s1+ " & " +s2+ " is "
+concatString);
}
public static void main(String[] args) {
Scanner sc = new Scanner(System.in);
System.out.println("Enter the following instances: ");
System.out.print("num(int): ");
int num = sc.nextInt();
System.out.print("a(int): ");
int a = sc.nextInt();
System.out.print("b(int): ");
int b = sc.nextInt();
System.out.print("s1(String): ");
String s1 = sc.next();
System.out.print("s2(String): ");
String s2 = sc.next();
OverLoad obj1 = new OverLoad();
obj1.num_calc(num);
obj1.num_calc(a, b);
obj1.num_calc(s1, s2);
}
}
```

11. Write a program that uses the function power() to raise a number m to power n. The function takes integer values for m and n and returns the result correctly. Use a default value of 2 for n to make the function calculate squares when this argument is omitted. Write a function main() to pass the value of m and n and display the calculated result.

Answer:

```
import java.util.Scanner;
public class PowerFunction {
int m, n;
PowerFunction(int m, int n) {
this.m = m;
this.n = n;
}
PowerFunction(int m) {
this.m = m;
this.n = 2;
}
public int power() {
int power = (int) Math.pow(this.m, this.n);
return power;
}
public static void main(String[] args) {
Scanner sc = new Scanner(System.in);
System.out.println("Enter the following credentials: ");
System.out.print("m: ");
int m = sc.nextInt();
System.out.print("n: ");
int n = sc.nextInt();
System.out.println("*************************Printing**************
***********");
PowerFunction obj1 = new PowerFunction(m, n);
int power = obj1.power();
System.out.println("The result of 'n raised to power of m' is " + power);
System.out.println("================================================
===");
PowerFunction obj2 = new PowerFunction(m);
power = obj2.power();
System.out.println("The result of 'default n raised to power of m' is " +
power);}
}
```