JADAVPUR UNIVERSITY

ADVANCED PROGRAMMING (JAVA AND PYTHON)
LAB ASSIGNMENTS

Name: SHRUTI PATHAK

Roll no: 002210503021

Dept: Computer Science and Engineering

Class: MCA 1st year 2nd sem

Session: 2022-2024

JAVA ASSIGNMENT 1:

1. **Write a program to accept two short integers from user and display the sum.**

SOURCE CODE:

```
import java.util.Scanner;

public class ass1_01 {
    public static void main(String args[]) {
        short num1, num2;
        int sum;
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter First number: ");
        num1 = sc.nextShort();
        System.out.println("Enter Second number: ");
        num2 = sc.nextShort();
        sc.close();
        sum = num1 + num2;
        System.out.println("Sum of these numbers: " + sum);
    }
}
```

output:

```
PS C:\Users\Shruti Pathak\Documents\MCA 2ND SEM\JAVA\ASS1> javac ass1_01.java
PS C:\Users\Shruti Pathak\Documents\MCA 2ND SEM\JAVA\ASS1> java ass1_01
Enter First number:
10
Enter Second number:
20
Sum of these numbers: 30
```

2. **Write a program that accepts number of command line parameters and displays the parameters and count of such parameters.**

SOURCE CODE:

```
public class ass1_02 {
    public static void main(String args[]) {
        if (args.length == 0)
            System.out.println("No argument passed!!!");
        else {

            System.out.println("The number of command line argument: " + args.length);
            System.out.println("The command line parameters: ");
            for (int i = 0; i < args.length; i++) {
                System.out.println(args[i]);
            }
        }
```

```
    }
}
```

output:

```
PS C:\Users\Shruti Pathak\Documents\MCA 2ND SEM\JAVA\ASS1> javac ass1_02.java
PS C:\Users\Shruti Pathak\Documents\MCA 2ND SEM\JAVA\ASS1> java ass1_02
No argument passed!!!

PS C:\Users\Shruti Pathak\Documents\MCA 2ND SEM\JAVA\ASS1> javac ass1_02.java
PS C:\Users\Shruti Pathak\Documents\MCA 2ND SEM\JAVA\ASS1> java ass1_02 A B C
The number of command line argument: 3
The command line parameters:
A
B
C
```

3. **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.**

SOURCE CODE:

```
import java.util.Scanner;
public class ass1_03 {
    public static void main(String args[]) {
        int cm;
        double inch,foot;
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter height in cm: ");
        cm=sc.nextInt();
        inch=cm/2.54;
        foot=cm/30.5;
        sc.close();
        System.out.println("The height in inch: "+inch);
        System.out.println("The height in foot: "+foot);

    }
}
```

output:

```
PS C:\Users\Shruti Pathak\Documents\MCA 2ND SEM\JAVA\ASS1> javac ass1_03.java
PS C:\Users\Shruti Pathak\Documents\MCA 2ND SEM\JAVA\ASS1> java ass1_03
Enter height in cm:
20
The height in inch: 7.874015748031496
The height in foot: 0.6557377049180327
```

4. **Write a program that accepts radius of a circle and displays area of the circle. Declare a constant pi equals to 3.14.**

SOURCE CODE:

```
import java.util.Scanner;
public class ass1_04 {
    public static void main(String args[]) {
        final double pi=3.14;
        double r,area;
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter radius of the circle: ");
        r=sc.nextDouble();
        area=pi*r*r;
        sc.close();
        System.out.println("Area of the circle is: "+area);

    }
}
```

output:

```
PS C:\Users\Shruti Pathak\Documents\MCA 2ND SEM\JAVA\ASS1> javac ass1_04.java
PS C:\Users\Shruti Pathak\Documents\MCA 2ND SEM\JAVA\ASS1> java ass1_04
Enter radius of the circle:
5
Area of the circle is: 78.5
```

5. **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.**

SOURCE CODE:

```
import java.util.Scanner;
public class ass1_05 {
    public static void main(String args[]) {
        String s1,s2;
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter a string: ");
        s1=sc.nextLine();
        s2=s1;
        System.out.println(s1==s2); //true
        System.out.println(s1.equals(s2)); //true

        System.out.println("Enter first string: ");
        s1=sc.nextLine();
        System.out.println("Enter second string(same as first): ");
        s2=sc.nextLine();
        System.out.println(s1==s2); //false because it compares the address of the strings
        System.out.println(s1.equals(s2)); //true because it compares the content
        sc.close();

    }

}
```

output:

```
PS C:\Users\Shruti Pathak\Documents\MCA 2ND SEM\JAVA\ASS1> javac ass1_05.java
PS C:\Users\Shruti Pathak\Documents\MCA 2ND SEM\JAVA\ASS1> java ass1_05
Enter a string:
shruti
true
true
Enter first string:
abc
Enter second string(same as first):
abc
false
true
PS C:\Users\Shruti Pathak\Documents\MCA 2ND SEM\JAVA\ASS1> java ass1_05
Enter a string:
mno
true
true
Enter first string:
xyz
Enter second string(same as first):
stu
false
false
```

6. **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.**

SOURCE CODE:

```java
import java.util.Scanner;
public class ass1_06 {
    static void show(int n)
    {
        System.out.println(n);
    }
    static void show(double n)
    {
        System.out.println(n);
    }
    public static void main(String args[]) {
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter a short value: ");
        short a=sc.nextShort();
        System.out.println("Enter a double value: ");
        double b=sc.nextDouble();
        show(a);
        show(b);
        sc.close();
```

```
    }

}
```

output:

```
PS C:\Users\Shruti Pathak\Documents\MCA 2ND SEM\JAVA\ASS1> javac ass1_06.java
PS C:\Users\Shruti Pathak\Documents\MCA 2ND SEM\JAVA\ASS1> java ass1_06
Enter a short value:
10
Enter a double value:
25.5
10
25.5
```

7. **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.**

SOURCE CODE:

```java
public class ass1_07 {
    static class Student{
        int roll;
        String name;
        double score;

        public void set(int r,String n,double s){
            roll=r;
            name=n;
            score=s;

        }

        public void display(){
            System.out.println("Student details: ");
            System.out.println("Roll: "+roll);
            System.out.println("Name: "+name);
            System.out.println("Score: "+score);
            System.out.println("");
        }

        public void copy(Student s){
            s.roll=roll;
            s.name=name;
            s.score=score;
        }
    }
```

```
    public static void main(String args[]) {
        Student s= new Student();
        Student s1 = new Student();
        s.set(21, "Shruti Pathak", 50);
        s.display();
        s.copy(s1);
        System.out.println("After copy: ");
        s1.display();
    }
}
```

output:

```
PS C:\Users\Shruti Pathak\Documents\MCA 2ND SEM\JAVA\ASS1> javac ass1_07.java
PS C:\Users\Shruti Pathak\Documents\MCA 2ND SEM\JAVA\ASS1> java ass1_07
Student details:
Roll: 21
Name: Shruti Pathak
Score: 50.0

After copy:
Student details:
Roll: 21
Name: Shruti Pathak
Score: 50.0
```

8. **Add constructors in the Student class of earlier problem so that objects can be created with i) roll only, ii) roll and name only, ii) 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 coy or not.**

SOURCE CODE:

```
public class ass1_08 {
    static class Student{
        int roll;
        String name;
        double score;
        public Student()
        {

        }
        public Student(int r)
        {
            roll=r;
        }
        public Student(int r,String n)
        {
            roll=r;
            name=n;
        }
        public Student(int r,String n,double s)
```

```
        {
            roll=r;
            name=n;
            score=s;
        }
        public Student(Student st)
        {
            roll=st.roll;
            name=st.name;
            score=st.score;
        }
        public void display(){
            System.out.println("Student details: ");
            System.out.println("Roll: "+roll);
            System.out.println("Name: "+name);
            System.out.println("Score: "+score);
            System.out.println("");


        }
    }

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Student s= new Student();
        System.out.println("Using constructor with no value: ");
        s.display();

        Student s1 = new Student(1);
        System.out.println("Using constructor with roll only: ");
        s1.display();

        Student s2 = new Student(2,"A");
        System.out.println("Using constructor with roll and name only: ");
        s2.display();

        Student s3 = new Student(3,"B",40.5);
        System.out.println("Using constructor with roll, name and score: ");
        s3.display();

        Student s4 = new Student(s3);
        System.out.println("Using copy constructor: ");
        s4.display();
        System.out.println("So,copy constructor works into deep copy ");

    }

}
```

output:

```
PS C:\Users\Shruti Pathak\Documents\MCA 2ND SEM\JAVA\ASS1> javac ass1_08.java
PS C:\Users\Shruti Pathak\Documents\MCA 2ND SEM\JAVA\ASS1> java ass1_08
Using constructor with no value:
Student details:
Roll: 0
```

Name: null
Score: 0.0

Using constructor with roll only:
Student details:
Roll: 1
Name: null
Score: 0.0

Using constructor with roll and name only:
Student details:
Roll: 2
Name: A
Score: 0.0

Using constructor with roll, name and score:
Student details:
Roll: 3
Name: B
Score: 40.5

Using copy constructor:
Student details:
Roll: 3
Name: B
Score: 40.5

So,copy constructor works into deep copy

9. **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.**

SOURCE CODE:

```
public class ass1_09 {
    static class BankAcct
    {
        int acn;
        double bal;
        static double rate;

        public static double initialize_rate(double r) {
            rate=r;
            return r;
        }
        public static double change_rate(double r) {
            rate=r;
            return r;
```

```
        }
        public static void display_rate() {
            System.out.println("\nThe interest rate is: "+rate);
        }
    }
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        double a=BankAcct.initialize_rate(5.5);
        BankAcct.display_rate();
        BankAcct b1=new BankAcct();
        b1.acn=100;
        b1.bal=10000;
        double i=b1.bal*a/100;
        System.out.println("The interest is: "+i);
        a=BankAcct.change_rate(6.5);
        BankAcct.display_rate();
        i=b1.bal*a/100;
        System.out.println("After changing the interest rate, interest is: "+i);

    }

}
```

output:

PS C:\Users\Shruti Pathak\Documents\MCA 2ND SEM\JAVA\ASS1> javac ass1_09.java
PS C:\Users\Shruti Pathak\Documents\MCA 2ND SEM\JAVA\ASS1> java ass1_09

The interest rate is: 5.5
The interest is: 550.0

The interest rate is: 6.5
After changing the interest rate, interest is: 650.0

10. 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.

SOURCE CODE:

```
import java.util.Scanner;
public class ass1_10 {
    static class Metric{
        double km,mile;
        public double km_to_mile(double km) {
            return km/1.5;
        }
        public double mile_to_km(double m) {
            return m*1.5;
        }
```

```
    }

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Metric d=new Metric();
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter distance in Kilometer: ");
        d.km=sc.nextDouble();
        System.out.println("Distance in Miles: "+d.km_to_mile(d.km));
        System.out.println("Enter distance in mile: ");
        d.mile=sc.nextDouble();
        System.out.println("Distance in kilometer: "+d.mile_to_km(d.mile));
        sc.close();
    }

}
```

output:

```
PS C:\Users\Shruti Pathak\Documents\MCA 2ND SEM\JAVA\ASS1> javac ass1_10.java
PS C:\Users\Shruti Pathak\Documents\MCA 2ND SEM\JAVA\ASS1> java ass1_10
Enter distance in Kilometer:
5
Distance in Miles: 3.3333333333333335
Enter distance in mile:
10
Distance in kilometer: 15.0
```

11. 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.

SOURCE CODE:

```
public class ass1_11 {
    static class instructor{
        String name;
        int phone;
        public void set(String n,int p)
        {
            name=n;
            phone=p;
        }
        public void view()
        {
            System.out.println("Instructor details: ");
            System.out.println("Name: "+name);
            System.out.println("phone: "+phone);
            System.out.println("");
```

```
        }
    }
    static class textbook{
        String title,author,publisher;
        public void set(String t,String a,String p)
        {
            title=t;
            author=a;
            publisher=p;
        }
        public void view()
        {
            System.out.println("Text book details: ");
            System.out.println("Title: "+title);
            System.out.println("Author name: "+author);
            System.out.println("Publisher: "+publisher);
            System.out.println("");

        }
    }
    static class course{
        String course_name;
        instructor ins;
        textbook book;
        public void set(String c,instructor i,textbook t)
        {
            course_name=c;
            ins=i;
            book=t;
        }
        public void view()
        {
            System.out.println("Course details: ");
            System.out.println("Course name: "+course_name);
            ins.view();
            book.view();
            System.out.println("");

        }
    }

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        instructor i=new instructor();
        i.set("Shruti", 893556671);
        i.view();
        textbook t=new textbook();
        t.set("JAVA","ABC","XYZ");
        t.view();
        course c=new course();
        c.set("JAVA Course", i, t);
        c.view();
    }

}
```

output:

```
PS C:\Users\Shruti Pathak\Documents\MCA 2ND SEM\JAVA\ASS1> javac ass1_11.java
PS C:\Users\Shruti Pathak\Documents\MCA 2ND SEM\JAVA\ASS1> java ass1_11
Instructor details:
Name: Shruti
phone: 893556671

Text book details:
Title: JAVA
Author name: ABC
Publisher: XYZ

Course details:
Course name: JAVA Course
Instructor details:
Name: Shruti
phone: 893556671

Text book details:
Title: JAVA
Author name: ABC
Publisher: XYZ
```

JAVA ASSIGNMENT 2:

1. Each customer of a bank has customer id, name, and current loan amount and phone number. One can change the attributes like name, phone number. A customer may ask for loan of certain amount. It is granted provided the sum of current loan amount and asked amount does not exceed credit limit (fixed amount for all customer). A customer may be a privileged amount. For such customers credit limit is higher. Once a loan is sanctioned necessary updates should be made. Any type of customer should be able to find his credit limit, current loan amount and amount of loan he can seek.

Design and implement the classes.

SOURCE CODE:

```
import java.util.Scanner;

public class ass2_01 {
    int cid;
    String cname;
    double cloan;
    int cphone;
    static double credit;
    static Scanner sc=new Scanner(System.in);

    void set()
    {
        System.out.println("Enter customer id: ");
        cid=sc.nextInt();
        System.out.println("Enter customer name: ");
        cname=sc.next();
        System.out.println("Enter customer phone no: ");
        cphone=sc.nextInt();
        System.out.println("Enter customer current loan amount: ");
        cloan=sc.nextDouble();
    }
    void view()
    {
        System.out.println("Customer Details:");
        System.out.println("Customer id: "+cid);
        System.out.println("Customer name: "+cname);
        System.out.println("Customer phone no: "+cphone);
        System.out.println("Customer current loan amount: "+cloan);

    }
    void set_credit()
    {
        System.out.println("Enter the credit limit: ");
        credit=sc.nextDouble();
    }
    void name_change()
    {
        System.out.println("Enter new customer name: ");
        cname=sc.next();
    }
```

```
void phoneno_change()
{
    System.out.println("Enter new customer phone no: ");
    cphone=sc.nextInt();
}
void ask_loan()
{
    System.out.println("Enter the loan amont: ");
    double loan=sc.nextDouble();
    if((loan+cloan)>credit)
        System.out.println("Exceed credit limit, loan cannot be sanctioned!!!");
    else
    {
        System.out.println("Loan is sanctioned!!!");
        cloan=cloan+loan;
        System.out.println("Your total loan amount is: "+cloan);
    }
}
void check_credit()
{
    System.out.println("Your credit value is: "+credit);
    System.out.println("Your current loan amount is: "+cloan);
    if(cloan>=credit)
        System.out.println("Credit limit reached, You can not seek more amount of loan!!!");
    else
        System.out.println("Amount of loan, you can seek: "+(credit-cloan));
}



public static void main(String[] args) {
    // TODO Auto-generated method stub
    ass2_01 cust1=new ass2_01();
    cust1.set_credit();
    cust1.set();
    cust1.view();
    System.out.println("Do you want to change the name of the customer(TRUE or FALSE)?");
    boolean ch=sc.nextBoolean();
    if(ch)
        cust1.name_change();
    System.out.println("Do you want to change the phone no of the customer(TRUE or FALSE)?");
    ch=sc.nextBoolean();
    if(ch)
        cust1.phoneno_change();
    cust1.ask_loan();
    cust1.check_credit();

    System.out.println("\nFor privilleged customer!");
    privilleged cust2=new privilleged();
    cust2.set_credit();
    cust2.set();
    cust2.view();
    System.out.println("Do you want to change the name of the customer(TRUE or FALSE)?");
    ch=sc.nextBoolean();
    if(ch)
        cust2.name_change();
    System.out.println("Do you want to change the phone no of the customer(TRUE or FALSE)?");
    ch=sc.nextBoolean();
```

```
        if(ch)
            cust2.phoneno_change();
        cust2.ask_loan();
        cust2.check_credit();

    }

}
class privilleged extends ass2_01
{
    static double credit;
    void set_credit()
    {
        System.out.println("Enter the credit limit for privilleged customer: ");
        credit=sc.nextDouble();
    }
    void ask_loan()
    {
        System.out.println("Enter the loan amont: ");
        double loan=sc.nextDouble();
        if((loan+cloan)>credit)
            System.out.println("Exceed credit limit, loan cannot be sanctioned!!!");
        else
        {
            System.out.println("Loan is sanctioned!!!");
            cloan=cloan+loan;
            System.out.println("Your total loan amount is: "+cloan);
        }
    }
    void check_credit()
    {
        System.out.println("Your credit value is: "+credit);
        System.out.println("Your current loan amount is: "+cloan);
        if(cloan>=credit)
            System.out.println("Credit limit reached, You can not seek more amount of loan!!!");
        else
            System.out.println("Amount of loan, you can seek: "+(credit-cloan));
    }
}
```

output:

```
PS C:\Users\Shruti Pathak\Documents\MCA 2ND SEM\JAVA\ASS2> javac ass2_01.java
PS C:\Users\Shruti Pathak\Documents\MCA 2ND SEM\JAVA\ASS2> java ass2_01
Enter the credit limit:
1000
Enter customer id:
101
Enter customer name:
shruti
Enter customer phone no:
988676980
Enter customer current loan amount:
0
Customer Details:
```

Customer id: 101
Customer name: shruti
Customer phone no: 988676980
Customer current loan amount: 0.0
Do you want to change the name of the customer(TRUE or FALSE)?
true
Enter new customer name:
Annesa
Do you want to change the phone no of the customer(TRUE or FALSE)?
false
Enter the loan amont:
500
Loan is sanctioned!!!
Your total loan amount is: 500.0
Your credit value is: 1000.0
Your current loan amount is: 500.0
Amount of loan, you can seek: 500.0

For privilleged customer!
Enter the credit limit for privilleged customer:
5000
Enter customer id:
501
Enter customer name:
Puja
Enter customer phone no:
987563548
Enter customer current loan amount:
4000
Customer Details:
Customer id: 501
Customer name: Puja
Customer phone no: 987563548
Customer current loan amount: 4000.0
Do you want to change the name of the customer(TRUE or FALSE)?
false
Do you want to change the phone no of the customer(TRUE or FALSE)?
true
Enter new customer phone no:
979854678
Enter the loan amont:
2000
Exceed credit limit, loan cannot be sanctioned!!!
Your credit value is: 5000.0
Your current loan amount is: 4000.0
Amount of loan, you can seek: 1000.0

2. For every person in an institute details like name, address (consists of premises number, street, city, pin and state), phone number, e-mail id are maintained. A person is either a student or a faculty. For student roll number and course of study are also

be maintained. For faculty employee id, department and specialisation are to be stored. One should be able to view the object details and set the attributes. For address, one may change it partially depending on the choice. Design and implement the classes.

SOURCE CODE:

```
import java.util.Scanner;

public class ass2_02 {
    String name;
    int pno;
    String street;
    String city;
    int pin;
    String state;
    int phone;
    String email;

    Scanner sc=new Scanner(System.in);
    void set()
    {
        System.out.println("Enter name: ");
        name=sc.next();
        System.out.println("Enter Address: ");
        System.out.println("Enter premises no: ");
        pno=sc.nextInt();
        System.out.println("Enter street name: ");
        street=sc.next();
        System.out.println("Enter city: ");
        city=sc.next();
        System.out.println("Enter pin no: ");
        pin=sc.nextInt();
        System.out.println("Enter state: ");
        state=sc.next();
        System.out.println("Enter phone no: ");
        phone=sc.nextInt();
        System.out.println("Enter email id: ");
        email=sc.next();
    }

    void view()
    {
        System.out.println("Name: "+name);
        System.out.println("Address: ");
        System.out.println("Premises no: "+pno);
        System.out.println("Street: "+street);
        System.out.println("City: "+city);
        System.out.println("Pin no: "+pin);
        System.out.println("State: "+state);
        System.out.println("phone no: "+phone);
        System.out.println("email: "+email);

    }
    void change_address()
    {
```

```
        System.out.println("Do you want to change the address?(TRUE or FALSE): ");
        boolean ch=sc.nextBoolean();
        if(ch)
        {
            System.out.println("Do you want to change the premises no?(TRUE or FALSE): ");
            ch=sc.nextBoolean();
            if(ch)
            {
                System.out.println("Enter new premises no: ");
                pno=sc.nextInt();
            }
            System.out.println("Do you want to change the street name?(TRUE or FALSE): ");
            ch=sc.nextBoolean();
            if(ch)
            {
                System.out.println("Enter new street name: ");
                street=sc.next();
            }
            System.out.println("Do you want to change the city name?(TRUE or FALSE): ");
            ch=sc.nextBoolean();
            if(ch)
            {
                System.out.println("Enter new city name: ");
                city=sc.next();
            }
            System.out.println("Do you want to change the pin no?(TRUE or FALSE): ");
            ch=sc.nextBoolean();
            if(ch)
            {
                System.out.println("Enter new pin no: ");
                pin=sc.nextInt();
            }
            System.out.println("Do you want to change the state name?(TRUE or FALSE): ");
            ch=sc.nextBoolean();
            if(ch)
            {
                System.out.println("Enter new state name: ");
                state=sc.next();
            }
            System.out.println("After changing new address is : ");
            System.out.println("Premises no: "+pno);
            System.out.println("Street: "+street);
            System.out.println("City: "+city);
            System.out.println("Pin no: "+pin);
            System.out.println("State: "+state);
        }
    }


    public static void main(String[] args) {
        // TODO Auto-generated method stub
        student s=new student();
        s.set_s();
        s.set();
        s.view_s();
        s.view();
        s.change_address();
```

```
        faculty f=new faculty();
        f.set_f();
        f.set();
        f.view_f();
        f.view();
        f.change_address();

    }

}
class student extends ass2_02
{
    int roll;
    String course;
    void set_s()
    {
        System.out.println("Enter student roll no: ");
        roll=sc.nextInt();
        System.out.println("Enter course of study: ");
        course=sc.next();
    }
    void view_s()
    {
        System.out.println("\nStudent Detils: ");
        System.out.println("Roll no: "+roll);
        System.out.println("Course of study: "+course);
    }
}
class faculty extends ass2_02
{
    int eid;
    String dept;
    String spec;
    void set_f()
    {
        System.out.println("\nEnter employee id: ");
        eid=sc.nextInt();
        System.out.println("Enter department: ");
        dept=sc.next();
        System.out.println("Enter speciallisation: ");
        spec=sc.next();
    }
    void view_f()
    {
        System.out.println("\nFaculty Employee Detils: ");
        System.out.println("Employee id: "+eid);
        System.out.println("Department: "+dept);
        System.out.println("Speciallisation: "+spec);
    }
}
```

output:

```
PS C:\Users\Shruti Pathak\Documents\MCA 2ND SEM\JAVA\ASS2> javac ass2_02.java
PS C:\Users\Shruti Pathak\Documents\MCA 2ND SEM\JAVA\ASS2> java ass2_02
Enter student roll no:
21
```

Enter course of study:
MCA
Enter name:
Shruti
Enter Address:
Enter premises no:
54
Enter street name:
Chanditala
Enter city:
kolkata
Enter pin no:
700054
Enter state:
Bengal
Enter phone no:
987668098
Enter email id:
shruti@gmail.com

Student Detils:
Roll no: 21
Course of study: MCA
Name: Shruti
Address:
Premises no: 54
Street: Chanditala
City: kolkata
Pin no: 700054
State: Bengal
phone no: 987668098
email: shruti@gmail.com
Do you want to change the address?(TRUE or FALSE):
true
Do you want to change the premises no?(TRUE or FALSE):
true
Enter new premises no:
37
Do you want to change the street name?(TRUE or FALSE):
false
Do you want to change the city name?(TRUE or FALSE):
true
Enter new city name:
Purulia
Do you want to change the pin no?(TRUE or FALSE):
true
Enter new pin no:
723102
Do you want to change the state name?(TRUE or FALSE):
False

After changing new address is :
Premises no: 37
Street: Chanditala
City: Purulia
Pin no: 723102
State: Bengal

Enter employee id:
101
Enter department:
IT
Enter speciallisation:
AI
Enter name:
Ankita
Enter Address:
Enter premises no:
23
Enter street name:
Jadavpur
Enter city:
kolkata
Enter pin no:
700032
Enter state:
bengal
Enter phone no:
987645698
Enter email id:
ankita@gmail.com

Faculty Employee Detils:
Employee id: 101
Department: IT
Speciallisation: AI
Name: Ankita
Address:
Premises no: 23
Street: Jadavpur
City: kolkata
Pin no: 700032
State: bengal
phone no: 987645698
email: ankita@gmail.com
Do you want to change the address?(TRUE or FALSE):
False

5. Design a student class with roll, name and score. Support must be there to set the score. Score is non-negative and cannot exceed 100. For invalid score an exception has to be raised. User of set score method will decide about the measures to deal

with the exception.

SOURCE CODE:

```
import java.util.*;

class MyException extends Exception {
    public String toString() {
        return "Score is non-negative and cannot exceed 100";
    }
}

class Student {
    private int roll;
    private String name;
    private double score;

    public void setRoll(int roll) {
        this.roll = roll;
    }

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

    public void setScore(double score) throws MyException {
        if (score < 0 || score > 100) {
            throw new MyException();
        }
        this.score = score;
    }

    public int getRoll() {
        return this.roll;
    }

    public String getName() {
        return this.name;
    }

    public double getScore() {
        return this.score;
    }
};

class ass2_05 {
    public static void main(String args[]) {
        Student s1 = new Student();
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter student roll: ");
        int roll = sc.nextInt();
        s1.setRoll(roll);
        System.out.print("Enter student name: ");
        sc.nextLine();
        String name = sc.nextLine();
        s1.setName(name);
        System.out.print("Enter student's score: ");
```

```
        double score = sc.nextDouble();
        try {
            s1.setScore(score);
            System.out.println("Student Roll : " + s1.getRoll() + "\nStudent Name : " +
s1.getName()
                    + "\nStudent Score : " + s1.getScore());
        } catch (MyException e) {
            System.out.println("Exception : " + e);
        }
        sc.close();
    }
}
```

output:

```
PS C:\Users\Shruti Pathak\Documents\MCA 2ND SEM\JAVA\ASS2> javac ass2_05.java
PS C:\Users\Shruti Pathak\Documents\MCA 2ND SEM\JAVA\ASS2> java ass2_05
Enter student roll: 21
Enter student name: Shruti
Enter student's score: 50
Student Roll : 21
Student Name : Shruti
Student Score : 50.0
PS C:\Users\Shruti Pathak\Documents\MCA 2ND SEM\JAVA\ASS2> java ass2_05
Enter student roll: 21
Enter student name: shruti
Enter student's score: 130
Exception : Score is non-negative and cannot exceed 100
PS C:\Users\Shruti Pathak\Documents\MCA 2ND SEM\JAVA\ASS2> java ass2_05
Enter student roll: 21
Enter student name: Shruti
Enter student's score: -50
Exception : Score is non-negative and cannot exceed 100
```

6. Consider a wrapper class for a numeric basic type. Check the support for the following: conversion from i) basic type to object ii) object to basic type iii) basic type to String iv) String (holding numeric data) to numeric object v) object to String.

SOURCE CODE:

```
import java.util.Scanner;

public class ass2_06 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter a integer: ");
        int a = sc.nextInt();
        Integer i = Integer.valueOf(a);
        System.out.println("Basic type to object => " + i);
        int b = i.intValue();
        System.out.println("Object to basic type => " + b);
        String num = String.valueOf(b);
        System.out.println("Basic type to string => " + num);
```

```
        Integer newNum = Integer.parseInt(num);
        System.out.println("String (holding numeric data) to  numeric object => " + newNum);
        String newString = Integer.toString(b);
        System.out.println("Object to string => " + newString);
        sc.close();
    }
}
```

output:

```
PS C:\Users\Shruti Pathak\Documents\MCA 2ND SEM\JAVA\ASS2> javac ass2_06.java
PS C:\Users\Shruti Pathak\Documents\MCA 2ND SEM\JAVA\ASS2> java ass2_06
Enter a integer:
20
Basic type to object => 20
Object to basic type => 20
Basic type to string => 20
String (holding numeric data) to  numeric object => 20
Object to string => 20
```

7. Take a String input that contains multiple words. Do the following: i) number of times 'a' appears ii) number of times "and" appears iii) whether it starts with "The" or not iv) put the String into an array of characters v) display the tokens in the String (tokens are the substrings separated by space or @ or .)

SOURCE CODE:

```
import java.util.*;

public class ass2_7 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter a string: ");
        String inputString = sc.nextLine();
        char[] chars = inputString.toCharArray();
        int count = 0;
        for (char c : chars) {
            if (c == 'a')
                count++;
        }
        System.out.println("Count of \'a\' is " + count);
        int countOfA = 0;
        System.out.println("Enter delimeter::");
        String del = new Scanner(System.in).nextLine();
        String[] arr;
        if (del.equals(" ")) {
            arr = inputString.split(" ");
        } else {
            arr = inputString.split("@");
        }
        String s = "and";
        for (String str : arr) {
            System.out.println(str);
```

```
            if (str.compareTo(s) == 0)
                countOfA++;
        }
        System.out.println("Count of \'and\' in the input string " + countOfA);
        if (arr[0].equals("The"))
            System.out.println("Input string starts with \'the\'");
        else
            System.out.println("Input string does not starts with \'the\'");

    }
}
```

output:

PS C:\Users\Shruti Pathak\Documents\MCA 2ND SEM\JAVA\ASS2> javac ass2_07.java
PS C:\Users\Shruti Pathak\Documents\MCA 2ND SEM\JAVA\ASS2> java ass2_07
Enter a string:
The sun gives us heat and light for the environment and the growth of organisms.
Count of 'a' is 4
Enter delimeter::

The
sun
gives
us
heat
and
light
for
the
environment
and
the
growth
of
organisms.
Count of 'and' in the input string 2
Input string starts with 'the'

PYTHON ASSIGNMENT 1:

1. **Write a prime generator program using only primes and using python loops.**

SOURCE CODE:

```
min=int(input("Enter the lowest range: "))
max=int(input("Enter the upper range: "))
print("The Prime numbers in the range are: ")
for i in range(min,max+1):
    if i > 1:
        for j in range(2,i):
            if (i%j) == 0:
                break
        else:
            print(i)
```

output:

```
PS C:\Users\Shruti Pathak\Documents\MCA 2ND SEM\PYTHON> py ass1.py
Enter the lowest range: 10
Enter the upper range: 30
The Prime numbers in the range are:
11
13
17
19
23
29
```

2. **Write a discount coupon code using dictionary in Python with different rate coupons for each day of the week.**

SOURCE CODE:

```
import datetime
coupon_codes={
    "Monday": 0.10,
    "Tuesday": 0.15,
    "Wednesday": 0.20,
    "Thursday": 0.25,
    "Friday": 0.30,
    "Saturday": 0.35,
    "Sunday": 0.40,
}

current_day= datetime.datetime.now().strftime("%A")

if current_day in coupon_codes:
    discount_rate = coupon_codes[current_day]
    coupon_code= f"DISCOUNT{int(discount_rate * 100)}"
    print(f"Today is {current_day}, and the discount rate is {discount_rate *100}%")
    print(f"Use coupon code '{coupon_code}' at checkout to avail the discount!")
```

```
else:
    print("No discount for today")
```

output:

PS C:\Users\Shruti Pathak\Documents\MCA 2ND SEM\PYTHON> py ass2.py
Today is Saturday, and the discount rate is 35.0%
Use coupon code 'DISCOUNT35' at checkout to avail the discount!

3. **Print first 10 odd and even numbers using iterators and compress. You can use duck typing.**

SOURCE CODE:

```
from itertools import compress,count
numbers=count(1)
odd_pattern= [True,False]*10
odd_numbers= compress(numbers,odd_pattern)
print("First 10 odd numbers : ")
for _ in range(10):
    print(next(odd_numbers),end = " ")
print()

numbers=count(1)
even_pattern= [False,True]*10
even_numbers= compress(numbers,even_pattern)
print("First 10 even numbers : ")
for _ in range(10):
    print(next(even_numbers),end = " ")
print()
```

output:

PS C:\Users\Shruti Pathak\Documents\MCA 2ND SEM\PYTHON> py ass3.py
First 10 odd numbers :
1 3 5 7 9 11 13 15 17 19
First 10 even numbers :
2 4 6 8 10 12 14 16 18 20

4. **Write a regular expression to validate a phone number.**

SOURCE CODE:

```
import re
n=input('Enter Mobile number : ')
r=re.fullmatch('[6-9][0-9]{9}',n)
if r!=None:
    print('Valid Number')
else:
    print('Not a valid number')
```

output:

```
PS C:\Users\Shruti Pathak\Documents\MCA 2ND SEM\PYTHON> py ass4.py
Enter Mobile number : 7679532692
Valid Number
PS C:\Users\Shruti Pathak\Documents\MCA 2ND SEM\PYTHON> py ass4.py
Enter Mobile number : 234567
Not a valid number
PS C:\Users\Shruti Pathak\Documents\MCA 2ND SEM\PYTHON> py ass4.py
Enter Mobile number : oiuyft
Not a valid number
```

5. **Write first seven Fibinacci numbers using generator next function/ yield in python. Trace and memorize the function.**

SOURCE CODE:

```
def fibo_generator():
    a,b=0,1
    count=0
    while count < 7:
        yield a
        a,b=b,a+b
        count +=1

fib_gen= fibo_generator()

print("Fibonacci Sequence with 7 terms: ")
for i in range(7):
    fibo_num=next(fib_gen)
    print(fibo_num, end= " ")
```

output:

```
PS C:\Users\Shruti Pathak\Documents\MCA 2ND SEM\PYTHON> py ass5.py
Fibonacci Sequence with 7 terms:
0 1 1 2 3 5 8
```

6. Write a simple program which loops over a list of user data (tuples containing a username, email and age) and adds each user to a directory if the user is at least 16 years old. You do not need to store the age. Write a simple exception hierarchy which defines a different exception for each of these error conditions:

- **the username is not unique**
- **the age is not a positive integer**
- **the user is under 16**
- **the email address is not valid (a simple check for a username, the @ symbol and a domain name is sufficient)**

Raise these exceptions in your program where appropriate. Whenever an exception occurs, your program should move onto the next set of data in the list. Print a different error message for each different kind of exception.

SOURCE CODE:

```
userdata=[
    ["user1", "user1@gmail.com",12],
    ["user2", "user2@gmail.com",15],
    ["user3", "user3@gmail.com",19],
    ["user3", "user3@gmail.com",19],
    ["user4", "user4@gmail.com",30],
    ["user5", "user5@gmail.com",-21],
    ["user6", "user6@gmail.com",67],
    ["user7", "user7@gmail.com",18],
    ["user8", "user8gmail.com",19]
]
user_dict={}
for user in userdata:
    try:
        if(user[0] in user_dict.keys()):
            raise Exception("The username is not unique for: " +user[0])
        elif(user[2] <0):
            raise Exception("The age is not a positive integer for: "+user[0])
        elif(user[2]<16):
            raise Exception("The user is under 16 for: "+user[0])
        elif '@' not in user[1] or '.com' not in user[1]:
            raise Exception("The email address is not valid for: "+user[0])
        else:
            user_dict[user[0]]= user[1]
    except Exception as e:
        print("Exception occured!!!",str(e))
        continue
print("\nValid users are: ")
print(user_dict)
```

output:

```
PS C:\Users\Shruti Pathak\Documents\MCA 2ND SEM\PYTHON> py ass5.py
Fibonacci Sequence with 7 terms:
0 1 1 2 3 5 8
PS C:\Users\Shruti Pathak\Documents\MCA 2ND SEM\PYTHON> py ass6.py
Exception occured!!! The user is under 16 for: user1
Exception occured!!! The user is under 16 for: user2
Exception occured!!! The username is not unique for: user3
Exception occured!!! The age is not a positive integer for: user5
Exception occured!!! The email address is not valid for: user8

Valid users are:
{'user3': 'user3@gmail.com', 'user4': 'user4@gmail.com', 'user6': 'user6@gmail.com', 'user7':
'user7@gmail.com'}
```

8. Create a list of all the numbers up to N=50 which are multiples of five using anonymous function.

SOURCE CODE:

```python
# Create a list of all numbers up to N=50 that are multiples of five using an anonymous function
N = 50

# Use a list comprehension with an anonymous function
multiples_of_five = [num for num in range(1, N+1) if (lambda x: x % 5 == 0)(num)]

# Print the list of multiples of five
print(multiples_of_five)
```

output:

```
PS C:\Users\Shruti Pathak\Documents\MCA 2ND SEM\PYTHON> py ass8.py
[5, 10, 15, 20, 25, 30, 35, 40, 45, 50]
```

10. Filter out the odd squares using map, filter, list.

SOURCE CODE:

```python
import math
def is_odd(num):
    return math.sqrt(num) % 2 != 0

def square(num):
    return num * num

# List of numbers
numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

# Use map to square each number
squared_numbers = list(map(square, numbers))

# Use filter to keep only odd numbers
filtered_numbers = list(filter(is_odd, squared_numbers))

print(filtered_numbers)
```

output:

```
PS C:\Users\Shruti Pathak\Documents\MCA 2ND SEM\PYTHON> py ass10.py
[1, 9, 25, 49, 81]
```

11. Let's find all Pythagorean triples whose short sides are numbers smaller than 10. use filter and comprehension.

SOURCE CODE:

```python
# Define the function to check if a triple is a Pythagorean triple
def is_pythagorean_triple(triple):
```

```
    a, b, c = triple
    return a**2 + b**2 == c**2

# Generate all possible combinations of numbers smaller than 10
numbers = range(1, 10)

# Use filter and comprehension to find the Pythagorean triples
pythagorean_triples = [
    (a, b, c)
    for a in numbers
     for b in numbers
     for c in numbers
    if is_pythagorean_triple((a, b, c))
]

# Print the Pythagorean triples
for triple in pythagorean_triples:
    print(triple)
```

output:

```
PS C:\Users\Shruti Pathak\Documents\MCA 2ND SEM\PYTHON> py ass11.py
(3, 4, 5)
(4, 3, 5)
```

12.Enumerate the sequence of all lowercase ASCII letters, starting from 1, using enumerate.

SOURCE CODE:

```
list_chars=["a","b","c","d","e","f","g","h","i","j","k","l","m","n","o","p","q",
            "r","s","t","u","v","w","x","y","z"]

for ascii,char in enumerate(list_chars,97):
    print(char,"-->",ascii)
```

output:

```
PS C:\Users\Shruti Pathak\Documents\MCA 2ND SEM\PYTHON> py ass12.py
a --> 97
b --> 98
c --> 99
d --> 100
e --> 101
f --> 102
g --> 103
h --> 104
i --> 105
j --> 106
k --> 107
l --> 108
```

m --> 109
n --> 110
o --> 111
p --> 112
q --> 113
r --> 114
s --> 115
t --> 116
u --> 117
v --> 118
w --> 119
x --> 120
y --> 121
z --> 122