

ASSESSMENT 1

ABHIRUPA MITRA
17BCE0437

Q1. Write a program to demonstrate the knowledge of students in basic Java concepts. Eg., Write a program to read the First name and Last name of a person, his weight and height using command line arguments. Calculate the BMI Index which is defined as the individual's body mass divided by the square of their height.

```
/**
 * @author Abhirupa Mitra 17BCE0437
 * PROGRAM 1
 */
import java.io.*;

public class Prog1 {

    public static void main(String args[])throws IOException
    {
        BufferedReader br = new BufferedReader (new InputStreamReader (System.in));
        String firstName, lastName;
        double weight,height;
        System.out.println("Enter first name:");
        firstName=br.readLine();
        System.out.println("Enter last name:");
        lastName=br.readLine();
        System.out.println("Enter height: (in cm)");
        height=Double.parseDouble(br.readLine());
        System.out.println("Enter weight:");
        weight=Double.parseDouble(br.readLine());
        double bmi= weight/(height*height);
        if(bmi<18.5)
            System.out.println("Underweight");
        else if(bmi<=25)
            System.out.println("Normal(healthy weight)");
        else if(bmi<=30)
            System.out.println("Overweight");
        else
            System.out.println("Obese Class");
    }
}
```

OUTPUT:

```
run:
Enter first name:
Ankit
Enter last name:
Sinha
```

Enter height: (in cm)
1.4
Enter weight:
80
40.81632653061225
Obese Class

Enter first name:
Abhirupa
Enter last name:
Mitra
Enter height: (in cm)
2
Enter weight:
70
17.5
Underweight

Enter first name:
Eshanee
Enter last name:
Ghosh
Enter height: (in cm)
2.2
Enter weight:
70
14.46280991735537
Underweight

2. Write a program to demonstrate the knowledge of students in multidimensional arrays and looping constructs.

Eg., If there are 4 batches in BTech(IT) learning 'ITE2005' course, read the count of the slow learners (who have scored <25) in each batch. Tutors should be assigned in the ratio of 1:4 (For every 4 slow learners, there should be one tutor). Determine the number of tutors for each batch. Create a 2-D jagged array with 4 rows to store the count of slow learners in the 4 batches. The number of columns in each row should be equal to the number of groups formed for that particular batch (Eg., If there are 23 slow learners in a batch, then there should be 6 tutors and in the jagged array, the corresponding row should store 4, 4, 4, 4, 4,3). Use for-each loop to traverse the array and print the details. Also print the number of batches in which all tutors have exactly 4 students.

```
/**  
 * @author Abhirupa Mitra 17BCEO437  
 * PROGRAM 1  
 */
```

```
import java.io.*;
```

```
public class Prog2 {
```

```
    public static void main(String args[])throws IOException
```

```

{
    BufferedReader br = new BufferedReader (new InputStreamReader (System.in));
    int arr[ ][ ]=new int[4][ ];
    for (int i=0;i<4;i++)
    {
        System.out.print("Enter no. of slow learners:");
        int slow = Integer.parseInt(br.readLine());
        double n=slow/4.0;
        if ((int)n*4== slow)
        {
            arr[i]= new int[(int)n];
            for(int j=0;j<n;j++)
                arr[i][j]=4;
        }

        else
        {
            int j=0;
            arr[i]= new int[(int)(n)+1 ];
            for(j=0;j<n;j++)
                arr[i][j]=4;
            arr[i][--j]=slow%4;
        }
    }
    System.out.println("Resulting array:");
    int c=0;
    for(int i=0;i<4;i++)
    {
        int flag=0;
        for(int j=0;j<arr[i].length; j++)
        {
            System.out.print(arr[i][j]+"\\t");
            if(arr[i][j]!=4)
            {
                flag=-1;
            }
        }
        if(flag==0)
            c++;
        System.out.println("");
    }
    System.out.println("Batches in which all tutors have exactly 4 students:"+c);
}
}

```

OUTPUT:

```

Enter no. of slow learners:12
Enter no. of slow learners:13
Enter no. of slow learners:15
Enter no. of slow learners:16
Resulting array:
4      4      4
4      4      4      1
4      4      4      3

```

4 4 4 4

Batches in which all tutors have exactly 4 students:2

Enter no. of slow learners:10

Enter no. of slow learners:11

Enter no. of slow learners:12

Enter no. of slow learners:16

Resulting array:

4 4 2

4 4 3

4 4 4

4 4 4 4

Batches in which all tutors have exactly 4 students:2

Enter no. of slow learners:8

Enter no. of slow learners:12

Enter no. of slow learners:16

Enter no. of slow learners:24

Resulting array:

4 4

4 4 4

4 4 4 4

4 4 4 4 4 4

Batches in which all tutors have exactly 4 students:4

3.

Write a program to demonstrate the knowledge of students in String handling. Eg., Write a program to read a chemical equation and find out the count of the reactants and the products. Also display the count of the number of molecules of each reactant and product.

Eg., For the equation, $2\text{NaOH} + \text{H}_2\text{SO}_4 \rightarrow \text{Na}_2\text{SO}_4 + 2\text{H}_2\text{O}$, the O/P should be as follows.

Reactants are 2 moles of NaOH, 1 mole of H₂SO₄.

Products are 1 mole of Na₂SO₄ and 2 moles of H₂O.

```
/**
```

```
 * Write a program to demonstrate the knowledge of students in String handling.
```

```
 */
```

```
/**
```

```
 * @author Abhirupa Mitra 17BCEO437
```

```
 * PROGRAM 3
```

```
 */
```

```
import java.io.*;
```

```
public class Prog3 {
```

```
    public static void main(String args[])throws IOException  
    {
```

```
        BufferedReader br = new BufferedReader (new InputStreamReader (System.in));  
        System.out.println("Enter chemical equation:");  
        String eqn=br.readLine();
```

```

char sp;
int stp, i;
int l=eqn.length();
int shft=-1;
System.out.println("Reactants are:\n");
for(i=0;i<l;i++)
{

    char c=eqn.charAt(i);
    if(i!=0)
        c=eqn.charAt(i+1);

    stp=-1;
    if(true)
    {
        for(int j=i;j<l;j++)
        {
            sp=eqn.charAt(j);
            if(sp=='+' || sp=='-' || j==l-1)
            {
                stp=j;
                if(sp=='-')
                    shft=1;
                break;
            }
        }
        if(i!=0)
            System.out.println(c+ " moles of "+ eqn.substring(i+2,stp));
        else
            System.out.println(c+ " moles of "+ eqn.substring(i+1,stp));

    }
    System.out.println();
    i=stp;
    if(shft==1)
        break;
}

//Print out the products
i=i+3;
System.out.print("Character at:"+eqn.charAt(i));
int org=i;
// System.out.println("Character at product beginning:"+eqn.charAt(i));
System.out.println("Products are:\n");
for(j=i;j<l;j++)
{

    char c=eqn.charAt(i);
    System.out.println("Petuk:"+c);

    //      if(i!=org)
    //          c=eqn.charAt(i+1);

```

```

        stp=-1;
        if(true)
        {
            for(int j=i;j<l;j++)
            {
                sp=eqn.charAt(j);
                if(sp=='+' || sp=='-' || j==l-1)
                {
                    stp=j;
                    if(sp=='-')
                        shft=1;
                    break;
                }
            }
            if(i!=org)
                System.out.println(c+ " moles of "+ eqn.substring(i+1,stp+1)+"\n");
            else
                System.out.println(c+ " moles of "+ eqn.substring(i+1,stp)+"\n");
        }
        System.out.println();
        i=stp;
    }
}

```

OUTPUT

Enter chemical equation:
2H2O + 3Na2SO4 -> 2NaOH + 2H2SO4
 Reactants are:

2 moles of H2O

3 moles of NA2SO4

Products are:

2 moles of NAOH

2 moles of H2SO4

Enter chemical equation:
2H2 + O2 -> 2H2O
 Reactants are:

2 moles of H2

2 moles of O2

Products are:

2 moles of H₂O

```
/**
 * Write a program to demonstrate the knowledge of students in String handling.
 */

/**
 * @author Abhirupa Mitra 17BCEO437
 * PROGRAM 4
 */
```

4.

Write a program to demonstrate the knowledge of students in advanced concepts of Java

string handling.

Eg., (Bioinformatics: finding genes) Biologists use a sequence of letters A, C, T, and G to model a genome. A gene is a substring of a genome that starts after a triplet ATG and ends before a triplet TAG, TAA, or TGA. Furthermore, the length of a gene string is a multiple of 3 and the gene does not contain any of the triplets ATG, TAG, TAA, and TGA. Write a program that prompts the user to enter a genome and displays all genes in the genome. If no gene is found in the input sequence, displays no gene. Here are the sample runs:

```
import java.io.*;

public class Prog4 {

    public static void main(String args[])throws IOException
    {
        BufferedReader br = new BufferedReader (new InputStreamReader (System.in));
        System.out.println("Enter gene sequence:");
        String gen=br.readLine();
        int stop;
        int i;
        int l= gen.length();
        for(i=0;i<l-2;i++)
        {
            String atg = gen.substring(i,i+3);
            if(atg.equals("ATG"))
            {
                stop=-1;
                int j=i+3;
                for(j<l-2;j++)
                {
                    String g=gen.substring(j,j+3);
                    if(g.equals("TAG") || g.equals("TAA") || g.equals("TGA"))
```

```

        {
            stop=j;
            break;
        }
        if(g.equals("ATG"))
        {
            stop=-1;
            break;
        }
    }
    System.out.println("stop:"+stop);
    if(stop!=-1)
    {
        System.out.println("Genome:"+gen.substring(i+3,stop));
        i=stop;
    }
}
}
}
}
}

```

OUTPUT:

Enter gene sequence:
 ATGCAGGAACTAA
 Genome:CAGGAAC

Enter gene sequence:
 ATGCTTAAGTGAGATGTTGGTAA
 Genome:CT
 Genome:TTGG

Enter gene sequence:
 ATGTTGTGTATATTAG
 Genome:TTGTGTATAT

5.

Write a program to demonstrate the knowledge of students in working with classes and objects.

Eg., Create a class Film with string objects which stores name, language and lead_actor and category (action/drama/fiction/comedy). Also include an integer data member that stores the duration of the film. Include parameterized constructor, default constructor and accessory functions to film class. Film objects can be initialized either using a constructor or accessor functions. Create a class FilmMain that includes a main function. In the main function create a vector object that stores the information about the film as objects. Use the suitable methods of vector class to iterate the vector object to display the following

- a. The English film(s) that has Arnold as its lead actor and that runs for shortest duration.
- b. The Tamil film(s) with Rajini as lead actor.
- c. All the comedy movies.

```

import java.util.*;
class Prog5
{

```



```

String name,language,lead_actor,category;
int duration;
FilmMain()
{
    duration=0;
    name=null;
    language=null;
    lead_actor=null;
    category=null;
}
FilmMain(int d,String n,String l,String la,String c)
{
    duration=d;
    name=n;
    language=l;
    lead_actor=la;
    category=c;
}
public void setData(int d,String n,String l,String la,String c)
{
    duration=d;
    name=n;
    language=l;
    lead_actor=la;
    category=c;
}
}
class Movies
{
    public static void main(String[]args)
    {
        Scanner sc=new Scanner(System.in);
        Prog5[]f=new Prog5[10];
        int num;
        System.out.println("Enter the number of movies:");
        num=sc.nextInt();
        for(int i=0;i<num;i++)
        {
            System.out.println("Enter the duration:");
            int d=sc.nextInt();
            System.out.println("Enter the name:");
            String n=sc.next();
            System.out.println("Enter the language:");
            String l=sc.next();
            System.out.println("Enter the lead_actor:");
            String la=sc.next();
            System.out.println("Enter the category:");
            String c=sc.next();
            f[i]=new FilmMain(d,n,l,la,c);
        }
        int short_duration=f[0].duration,k=0;
        for(int i=0;i<num;i++)
        {
            if(f[i].lead_actor.equals("Arnold")&&f[i].language.equals("English"))
            {
                short_duration=f[i].duration;
            }
        }
    }
}

```

```

        for(int j=0;j<num;j++)
        {
            if(f[j].lead_actor.equals("Arnold")&&f[j].language.equals("English"))
            {
                if(f[j].duration<short_duration)
                {
                    short_duration=f[j].duration;
                    k=j;
                }
            }
        }
        break;
    }
}
System.out.println("The Arnold English movie of shortest duration is");
System.out.println(f[k].name);
System.out.println("The Rajini movies are");
for(int j=0;j<num;j++)
{
    if(f[j].lead_actor.equals("Rajini"))
    {
        System.out.println(f[j].name);
    }
}
System.out.println("The comedy movies are");
for(int j=0;j<num;j++)
{
    if(f[j].category.equals("Comedy"))
    {
        System.out.println(f[j].name);
    }
}
}
}

```

OUTPUT:

Enter the number of movies:

4

Enter the duration:

120

Enter the name:

Random Horses

Enter the language:

Englis

Enter the lead_actor:

Arnold

Enter the category:

Comedy

...

6. Basic Program

Read the following details of 'n' students using Scanner class methods and display the same.

- Registration number (String)
- Name (String that may contain first name, middle name and last name)

- CGPA (Floating point number)
- Programme Name(String)
- School Name (String with multiple words)
- Proctor Name (String that may contain first, middle and last names)

```
import java.util.*;
public class Prog6
{
    public static void main(String [] args)
    {
        Scanner sc=new Scanner(System.in);
        int choice;
        do
        {
            System.out.println("Registration number:");
            String regno=sc.next();
            System.out.println("Name:");
            String name=sc.next();
            System.out.println("CGPA:");
            float CGPA=sc.nextFloat();
            System.out.println("Program:");
            String pgrm=sc.next();
            System.out.println("School:");
            String school=sc.next();
            System.out.println("Proctor:");
            String proctor=sc.next();
            System.out.println("Registration number:"+regno);
            System.out.println("Name:"+name);
            System.out.println("CGPA:"+CGPA);
            System.out.println("Program:"+pgrm);
            System.out.println("School:"+school);
            System.out.println("Proctor:"+proctor);
            System.out.println("Do you want to enter and display details? 1-Yes/0-No");
            choice=sc.nextInt();
        }while(choice!=0);
    }
}
```

OUTPUT:

```
Registration number:
17BCE0437
Name:
Abhirupa
CGPA:
9.1
Program:
CSE
School:
SCOPE
Proctor:
Dr. Shashank Mouli Satapathy
Registration number:17BCE0437
```

Name:Abhirupa
CGPA:9.1
Program:CSE
School:SCOPE
Proctor:Shashank
Do you want to enter and display details? 1-Yes/0-No 0

7.The table mentioned above contains the course codes of courses that are to be offered in the next semester. A faculty can give 5 preferences from within the list for course allocation.

Get the preferences through command line arguments. If a faculty fails to give his/her preferences, then the Time table committee may fill the preferences by choosing any 5 courses randomly from the list. If a faculty gives less than 5 preferences, the left out ones will be filled randomly. Write a program to implement the same and print the 5 preferences at last.

```
import java.util.Scanner;
import java.util.Random;
class Courses
{
String[][] c= {"ITA1001", "ITA1002", "ITA1003", "ITA1004", "ITA1005", "ITA1006"}, {"ITA2001",
"ITA2002", "ITA2003", "ITA2004", "ITA2005", "ITA2006"}, {"ITE1001", "ITE1002", "ITE1003",
"ITE1004", "ITE1005", "ITE1006"}, {"ITE2001", "ITE2002", "ITE2003", "ITE2004", "ITE2005",
"ITE2006"}, {"SWE1001", "SWE1002", "SWE1003", "SWE1004", "SWE1005", "SWE1006"},
{"SWE2001", "SWE2002", "SWE2003", "SWE2004", "SWE2005", "SWE2006"}};
public void courseDetails()
{

System.out.println("The courses offered are:");
System.out.println();
for(int i=0;i<6;i++)
{
for(int j=0;j<6;j++)
System.out.print(c[i][j]+" ");
System.out.println();
}
}
public void select()
{
int[] a=new int[6];
int r,c1,no,i;
Scanner sc=new Scanner(System.in);
System.out.println("For faculty: how many course you want to allocate:");
no=sc.nextInt();
System.out.println("Enter the number for courses:");
for(i=0;i<no;i++)
a[i]=sc.nextInt();
if(no<5)
{
System.out.println("The remaining courses will be selected by committee member randomly");
for(i=no;i<5;i++)
a[i]=(int)(Math.random()*35);
}
}
```

```

System.out.println("The five preferences are:");
for(i=0;i<5;i++)
{
r=a[i]/6;
c1=a[i]%6;
System.out.println(c[r][c1]);
}
}
}
}
public class Course
{
public static void main(String[] args)
{
Courses c1=new Courses();
c1.courseDetails();
c1.select();
}
}
}

```

8.. String and StringBuffer

A hash algorithm uses rotation and fold shift methods to compute the address at which the user input has to be stored. Define a static method to perform rotation of the data by moving the least significant digit to the most significant bit position. Also define a non-static method to perform fold shift by dividing the rotated data into segments of length 2 and then add all the segments to get the hash address. If the sum has more than 2 digits, delete the most significant digit and the resulting number is the address. Invoke these methods from main() method.

Eg., If the data is 112286, after rotation it should be 611228 and after fold shift it should be $61 + 12 + 28 = 101 = 01$ (after deleting the most significant digit)

```

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

```

```

/**
 *
 * @author abhirupa
 */

```

```

import java.io.*;
public class Prog8 {
    public static int rotation(int n){
        int d=n%10;
        int c=0;
        for(int i=n;i!=0;i/=10,c++); //Counting no. of digits
        int pw=1;
        for(int i=1;i<=c;i++,pw*=10);
        int p = pw*d +n;
        p=p/10;
        return p;
    }

    public void foldData(int n){

```

```

        int s=0;
        for(int i=n;i!=0;i/=100)
        {
            System.out.println("s:"+i%100);
            s+=(i%100);
        }
        String dm=Integer.toString(s);
        if(s/100 !=0)
        {
            dm=dm.substring(1,dm.length());
        }
        System.out.println(dm);
    }
    public static void main(String args[])throws IOException{
        BufferedReader br = new BufferedReader (new InputStreamReader (System.in));
        Prog8 obj = new Prog8();
        int n =Integer.parseInt(br.readLine());

        int n0 = obj.rotation(n);
        System.out.println("n is"+n0);
        obj.foldData(n0);

        // System.out.println("Running Instance");
    }
}

```

OUTPUT:

```

112286
n is611228
s:28
s:12
s:61
01

```

```

115464
n is411546
s:46
s:15
s:41
02

```

```

115421
n is111542
s:42
s:15
s:11
68

```

9. String and StringBuffer

Consider a Java program containing a statement to invoke `format()` method for displaying the output. Write a program to verify the syntax correctness of the statement by checking for the following.

- The number of format specifiers and arguments should match.
- Datatype of the arguments should match the format specifiers used.

```
import java.util.*;
import java.lang.*;
public class Prog9 {

    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        String[][] data = {"a","b","s","x"}, {"d","f","s","d"};
        System.out.println("Enter statement: ");
        String ip = scan.nextLine();
        String parts[] = ip.split("%");
        String parts2[] = ip.split(",");
        int i,j,k;
        char format[] = new char[10];
        char var[] = new char[10];
        for(i=0;i<parts.length;i++)
        {
            format[i]=parts[i].charAt(0);
        }
        for(i=0;i<parts2.length;i++)
        {
            var[i]=parts2[i].charAt(0);
        }
        int check=0;
        for(i=1;i<(parts.length-1);i++)
        {
            for(j=0;j<data[0].length;j++)
            {
                if(Character.toString(var[i])==data[0][j])
                {
                    System.out.println("variable was found here.");
                    if(Character.toString(format[i])==data[1][j])
                    {
                        System.out.println("Format was found here");
                        check=check+1;
                    }
                }
            }
        }
        if(check>var.length)
        {
            System.out.println("Syntax is wrong.");
        }
        else
        {
            System.out.println("Syntax is correct.");
        }
    }
}
```

```
}
```

```
}
```

OUTPUT:

Enter statement:

```
System.out.format("sum is %d"+" avg is %f ", a,b);
```

Syntax is correct.

Enter statement:

```
System.out.format(" name is %s"+"sum is %d avg is %f ", s,a,b);
```

Syntax is correct.