

16BIT0155  
PRANJAL KHANDELWAL  
JAVA LAB ASSIGNMENT – 6 to 10

**6)** Write a program to demonstrate the knowledge of students in creation of abstract classes and working with abstract methods.

Eg., Define an abstract class 'Themepark' and inherit 2 classes 'Queensland' and 'Wonderla' from the abstract class. In both the theme parks, the entrance fee for adults is Rs. 500 and for children it is Rs. 300. If a family buys 'n' adult tickets and 'm' children tickets, define a method in the abstract class to calculate the total cost. Also, declare an abstract method playGame() which must be redefined in the subclasses.

In Queensland, there are a total of 30 games. Hence create a Boolean array named 'Games' of size 30 which initially stores false values for all the elements. If the player enters any game code that has already been played, a warning message should be displayed and the user should be asked for another choice. In Wonderla, there are a total of 40 different games. Thus create an integer array with 40 elements. Here, the games can be replayed, until the user wants to quit. Finally display the total count of games that were repeated and count of the games which were not played at all.

```
package cat2lab;
import java.util.*;
abstract class Themepark
{
    int totalcost(int m,int n){
        return m*500+n*300;
    }
    abstract void playGame();
}
class Queensland extends Themepark
{
    boolean Games[]=new boolean [30];
    void playGame()
    {
        System.out.println("Welcome to Queensland");
        int n,m;
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter the no. of adults");
        m=sc.nextInt();
        System.out.println("Enter the no. of Children");
```

```

n= sc.nextInt();
    for(int i=0;i<m+n;++i)
    {
        System.out.print("Enter the game You would like to play");
        int in=sc.nextInt();
        if(Games[in]==false)
        {
            Games[in]=true;
            System.out.println("The user is playing the game " + in);
        }
        else
        {
            System.out.println("Warning Message: The user has already
played this game" + in);
            System.out.println(".....try Again.....");
            i--;
        }
    }
    System.out.print("the games which are not played: ");
    for(int i=0;i<30;++i)
    {
        if(Games[i]==false)
        {
            System.out.print(i+1+" ");
        }
    }
    int t=totalcost(m,n);
    System.out.println("\n");
    System.out.println("The total cost of the game:  "+t);
}
}
class Wonderla extends Themepark
{
    int[] a = new int[40];
    int [] b=new int [40];
    void playGame()
    {
        Scanner sc1=new Scanner(System.in);
        System.out.println("Enter the no. of adults");
        int m=sc1.nextInt();
        System.out.println("Enter the no. of Children");
        int n= sc1.nextInt();
        for(int i=0;i<40;++i)
        {
            a[i]=i;
        }
        for(int i=0;i<40;++i)
        {
            b[i]=0;
        }
        int ad=0;
        int ch=0;
        for(int i=1;i<m+n+1;++i)
        {

```

```

        int c=1;
        System.out.println("Enter U if the person is adult and C for
children");
        String t= sc1.next();
        System.out.println("Enter the input in b/w 0 to 40 for user " + i);
        int x=sc1.nextInt();
        if(t.equals("U"))
        {
            ad++;
        }
        else
        {
            ch++;
        }
        for(int j=0;j<40;++j)
        {
            if(x==a[j])
            {
                b[j]=c;
                c++;
            }
        }
        System.out.println("You want to play more 1 for yes 0 for no");
        int p= sc1.nextInt();
        while(p==1)
        {
            if(t.equals("U"))
                ad++;
            else
                ch++;
            System.out.println("Enter the input in b/w 0 to 40");
            x=sc1.nextInt();
            for(int j=0;j<40;++j)
            {
                if(x==a[j])
                {
                    b[j]=c;
                    c++;
                }
            }
            System.out.println("U want to play more 1 for yes 0 for no");
            p=sc1.nextInt();
        }
        if(p==0)
        {
            continue;
        }
    }
    int count=0;
    int count1=0;
    int e=0;

    for(int i=0;i<40;++i)
    {

```

```

        if(b[i]>0)
        {
            e=b[i];
            if(b[i]>1)
            count++;
            count1=count1+e;
        }
    }
    System.out.println("Total number of games that are played repeatedly " +
count);
    System.out.println("Total number of games that are played " + count1);
    for(int i=0;i<40;++i)
    {
        System.out.print(b[i]+" ");

    }
    System.out.println();
    int t=totalcost(ad,ch);
    System.out.println("\n");
    System.out.println("The total cost of the game: "+t);
}
public class Sixth {
    public static void main(String args[])
    {
        System.out.println("16BIT0155  Pranjali Khandelwal");
//        Queensland obj1=new Queensland();
//        obj1.playGame();
        Wonderla obj2 =new Wonderla();
        obj2.playGame();
    }
}

```

```

16BIT0155  Pranjali Khandelwal
Welcome to Queensland
Enter the no. of adults
1
Enter the no. of Children
1
Enter the game You would like to play3
The user is playing the game 3
Enter the game You would like to play4
The user is playing the game 4
the games which are not played: 1 2 3 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30

The total cost of the game: 800

```



[illegible]

7) Write a program to demonstrate the knowledge of students in Java Exception handling. Eg., Read the Register Number and Mobile Number of a student. If the Register Number does not contain exactly 9 characters or if the Mobile Number does not contain exactly 10 characters, throw an `IllegalArgumentException`. If the Mobile Number contains any character other than a digit, raise a `NumberFormatException`. If the Register Number contains any character other than digits and alphabets, throw a `NoSuchElementException`. If they are valid, print the message 'valid' else 'invalid'

```
package cat2lab;
import java.util.*;
public class Seventh {
    public static void main(String args[])
    {
        System.out.println("16BIT0155      PRANJAL KHANDELWAL");
        int c=0;
        Scanner sc =new Scanner(System.in);
```

```

String mobile=sc.next();
String str= sc.next();
if(str.length()!=9 || mobile.length()!=10)
{
    c=1;
    throw new IllegalArgumentException("Invalid");
}
char ch1[]=mobile.toCharArray();
char ch2[]=str.toCharArray();
for(int i=0;i<mobile.length();++i)
{
    if(Character.isDigit(ch1[i])!=true)
    {
        c=1;
        throw new NumberFormatException("Invalid");
    }
}
for(int i=0;i<str.length();++i)
{
    if(Character.isLetter(ch2[i])!=true && Character.isDigit(ch2[i])!=true)
    {
        c=1;
        throw new NumberFormatException("Invalid");
    }
}
if(c!=1)
{
    System.out.println("Valid Arguments");
}
}

```

## Output:

```

16BIT0155    PRANJAL KHANDELWAL
9829328117
16BIT0155
Valid Arguments

```

```
16BIT0155    PRANJAL KHANDELWAL
982
16BIT0155
Exception in thread "main" java.lang.IllegalArgumentException: Invalid
    at cat2lab.Seventh.main(Seventh.java:14)
```

```
16BIT0155    PRANJAL KHANDELWAL
1234567896
16BIT016%
Exception in thread "main" java.lang.NumberFormatException: Invalid
    at cat2lab.Seventh.main(Seventh.java:32)
```

8) Write a program to demonstrate the knowledge of students in working with user-defined packages and sub-packages. Eg., Within the package named 'primespackage', define a class Primes which includes a method checkForPrime() for checking if the given number is prime or not. Define another class named TwinPrimes outside of this package which will display all the pairs of prime numbers whose difference is 2. (Eg, within the range 1 to 10, all possible twin prime numbers are (3,5), (5,7)). The TwinPrimes class should make use of the checkForPrime() method in the Primes class.

#### Prime.java

```
package primespackage;

public class Prime {

    public static boolean checkPrime(int n) {
        int t;
        boolean ip = true;
```



```

        for (int j = 2; j <= n / 2; j++)
        {
            t = n % j;
            if (t == 0)
            {
                ip = false;
                break;
            }
        }

        return ip;
    }
}

```

## TwinsPrime.java

```

package primespackage;

import java.util.*;
import primespackage.Prime;

public class TwinsPrime {
    public static void main(String args[])
    {
        Scanner scan = new Scanner(System.in);
        System.out.print("Enter first number : ");
        int n1 = scan.nextInt();
        System.out.print("Enter second number : ");
        int n2 = scan.nextInt();

        System.out.println("Twin-prime numbers are: ");
        for(int j = n1; j <= n2; j++) {
            if(Prime.checkPrime(j) && Prime.checkPrime(j+2) && j > 1)
            {
                System.out.print("(" + j + "," + (j+2) + ")");
                System.out.print(" ");
            }
        }
    }
}

```

```
Enter first number : 1
Enter second number : 10
Twin-prime numbers are:
(3,5) (5,7)
```

```
Enter first number : 1
Enter second number : 100
Twin-prime numbers are:
(3,5) (5,7) (11,13) (17,19) (29,31) (41,43) (59,61) (71,73)
```

9) Write a program to demonstrate the knowledge of students in File handling. Eg., Define a class 'Donor' to store the below mentioned details of a blood donor. Name, age, Address, Contactnumber, bloodgroup, date of last donation Create 'n' objects of this class for all the regular donors at Vellore. Write these objects to a file. Read these objects from the file and display only those donors' details whose blood group is 'A+ve' and had not donated for the recent six months.

```
package primespackage;
```

```
import java.io.*;
import java.util.*;
import java.text.*;
```

```
public class bloodDonation implements Serializable {
    String name;
    String address;
    String bloodGroup;
    int age;
    long contactNumber;
    Date lastDonation;
```

```

    public bloodDonation() {
        // TODO Auto-generated constructor stub
    }
    public bloodDonation(String name,int age,String address,long
contactNumber,String bloodGroup,Date lastDonation) {
        this.name=name;
        this.age=age;
        this.address=address;
        this.contactNumber=contactNumber;
        this.bloodGroup=bloodGroup;
        this.lastDonation=lastDonation;
    }
    public static void main(String[] args) {
        Write obj=new Write();
        obj.writeToFile();
        Read obj1=new Read();
        obj1.readFile();
    }
}
class Write {
    void writeToFile(){

        bloodDonation s;

        Vector<bloodDonation> v = new Vector<>();
        v.add(new
bloodDonation("Pranjal",21,"Letsee",8860,"A+",dateClass.setDate(2020,0,26)));
        v.add(new
bloodDonation("Bhagyashree",21,"Mumbai",8860,"B+",dateClass.setDate(2019,0,26)));
        v.add(new bloodDonation("Heyya",20,"whokonws",8860,"AB-
",dateClass.setDate(2020,1,26)));
        v.add(new
bloodDonation("RAM",21,"Ayodhya",8860,"A+",dateClass.setDate(2018,1,27)));
        v.add(new
bloodDonation("Shayam",20,"jaipur",8860,"A+",dateClass.setDate(2020,0,26)));
        v.add(new
bloodDonation("Aayushi",21,"Kota",8860,"A+",dateClass.setDate(2020,0,26)));//first
object

        //creat file and write these object to a file
        File f = new File("objectTest.txt");
        try {
            FileOutputStream fos = new FileOutputStream(f);
            ObjectOutputStream oos = new ObjectOutputStream(fos);
            oos.writeObject(v);
            oos.close();
            System.out.println("data write successfully");

        } catch (FileNotFoundException ex) {
            ex.printStackTrace();
        } catch (IOException ex) {
            ex.printStackTrace();
        }
    }
}

```

```

    }
}

class Read{

    void readFile() {

        File f = new File("objectTest.txt");
        try {
            FileInputStream fis = new FileInputStream(f);
            ObjectInputStream ois = new ObjectInputStream(fis);

            //        ois.readObject();
            //        ois.close();

            Vector<bloodDonation> deserializeStudent =
            (Vector<bloodDonation>)ois.readObject();
            //System.out.println(deserializeStudent);
            ois.close();
            dateClass obj=new dateClass();
            obj.sixMonth();
            Iterator<bloodDonation> iter = deserializeStudent.iterator();
            while(iter.hasNext()){
                bloodDonation s = iter.next();
                if(dateClass.daysBetween(s.lastDonation)>180 &&
(s.bloodGroup).equals("A+")) {
                    System.out.println(dateClass.daysBetween(s.lastDonation));
                    System.out.println(
                        "Name: " + s.name );
                    System.out.println(
                        "Age: " + s.age
                    );
                    System.out.println(
                        "Blood Group: " + s.bloodGroup
                    );
                    System.out.println(
                        "Address: " + s.address
                    );
                    System.out.println(
                        "Mobile number: " + s.contactNumber
                    );
                    System.out.println(
                        "Last Donation: " + s.lastDonation
                    );
                    System.out.println(" ");
                    System.out.println(" ");
                    System.out.println(" ");
                }
            }

        } catch (FileNotFoundException ex) {
            ex.printStackTrace();
        }
    }
}

```

```

        } catch (IOException ex) {
            ex.printStackTrace();
        } catch (ClassNotFoundException ex) {
            ex.printStackTrace();
        }
    }
}

class dateClass{
    public static long daysBetween(Date one,Date two) {
        long difference=(one.getTime()-two.getTime())/86400000;
        return Math.abs(difference);
    }
    public static long daysBetween(Date one) {
        Date two=new Date();
        long difference=(one.getTime()-two.getTime())/86400000;
        return Math.abs(difference);
    }
    public static void sixMonth() {
        Date today=new Date();

        Calendar myNextCalender=Calendar.getInstance();
        myNextCalender.set(2020,0,26);
        Date nyd=myNextCalender.getTime();
        long day=daysBetween(today,nyd);
        //System.out.println(day);

    }
    public static Date setDate(int year,int month,int day) {
        Calendar myNextCalender=Calendar.getInstance();
        myNextCalender.set(year,month,day);
        Date nyd=myNextCalender.getTime();
        return nyd;
    }
}

```

data write successfully

331

Name: Pranjali

Age: 21

Blood Group: A+

Address: Letsee

Mobile number: 8860

Last Donation: Sun Jan 26 13:40:48 IST 2020

366

Name: RAM

Age: 21

Blood Group: A+

Address: Ayodhya

Mobile number: 8860

Last Donation: Tue Feb 27 13:40:48 IST 2018

331

Name: Shayam

Age: 20

Blood Group: A+

Address: jaipur

Mobile number: 8860

Last Donation: Sun Jan 26 13:40:48 IST 2020

331

Name: Aayushi

Age: 21

Blood Group: A+

Address: Kota

Mobile number: 8860

<

Mobile number: 8860

Last Donation: Sun Jan 26 13:40:48 IST 2020

**10) Write a program to demonstrate the knowledge of students in multithreading. Eg., Three students A, B and C of B.Tech-IT II year contest for the PR election. With the total strength of 240 students in II year, simulate the vote casting by generating 240 random numbers (1 for student A, 2 for B and 3 for C) and store them in an array. Create four threads to equally share the task of counting the number of votes cast for all the three candidates. Use synchronized method or synchronized block to update the three count variables. The main thread should receive the final vote count for all three contestants and hence decide the PR based on the values received.**

```
package cat2lab;
import java.util.*;
class Table{
    static int [] vote= new int[240];
    static int count1=0;
    static int count2=0;
    static int count3=0;
    void Voterand()
    {
        System.out.println("16BIT01555 PRANJAL KKHANDELWAL");
        for(int i=0;i<240;++i)
        {
            vote[i]=(int )(Math.random()*3+1);
        }
        for(int i=0;i<240;++i)
        {
            System.out.print(vote[i]+" ");
        }
        System.out.println(" ");
    }
    public synchronized static void count(int m,int n){
        for(int i=m;i<n;++i)
        {
            if(vote[i]==1)
            {
                count1++;
            }
            if(vote[i]==2)
            {
                count2++;
            }
            if(vote[i]==3)
            {
                count3++;
            }
        }
    }
}

public void display()
```

```

        {
            System.out.println(count1);
            System.out.println(count2);
            System.out.println(count3);
            System.out.println(count1+count2+count3);
        }
    }

    class MyThread1 extends Thread{
    public void run(){
        Table.count(0,60);
    }
    }

    class MyThread2 extends Thread{
    public void run(){
        Table.count(60,120);
    }
    }

    class MyThread3 extends Thread{
    public void run(){
        Table.count(120,180);
    }
    }
    class MyThread4 extends Thread{
    public void run(){
        Table.count(180,240);
    }
    }

    public class classB{
    public static void main(String t[]){
        Table obj1=new Table();
        obj1.Voterand();
        MyThread1 t1=new MyThread1();
        MyThread2 t2=new MyThread2();
        MyThread3 t3=new MyThread3();
        MyThread4 t4=new MyThread4();
        t1.start();
        try{
            t1.join();
        }catch(Exception e){System.out.println(e);}

        t2.start();
        try{
            t2.join();
        }catch(Exception e){System.out.println(e);}

        t3.start();

        try{
            t3.join();
        }catch(Exception e){System.out.println(e);}
        t4.start();
    }
    }

```



```
    obj1.display();  
}  
}
```

## OUTPUT:

```
16BIT01555  PRANJAL KKHANDELWAL  
1 1 3 2 2 3 3 3 3 2 2 1 1 3 1 1 2 2 2 3 2 2 3 1 1 3 3 3 1 3 1 2 1 3 1 2 3 2 1 2 1 3 3 1 2 2 1 1 2 2 1 3 3 1 2 2 2 2 2 1 3 2 2 :  
count of A 55  
Count of B 71  
Count of C 54  
240  
B wins
```

---

```
16BIT01555  PRANJAL KKHANDELWAL  
1 3 1 2 1 2 1 2 2 3 3 1 2 1 2 2 1 2 3 1 2 3 3 1 3 1 1 1 2 2 1 2 1 3 2 2 1 2 1 3 2 1 1 3 3 2 1 2 3 3 1 2 3 1 1 1 :  
count of A 61  
Count of B 65  
Count of C 54  
240  
B wins
```