

OBJECT ORIENTED PROGRAMMING WITH JAVA

PROBLEM SHEET-2

Ques.1

Create a package “p1”. Under this package create a class called “stud” which contains instance variables rollno, name and class. Extend class “result” from “stud” which contains instance variables marks in OS, marks in Java, marks in DBMS. Store class result in package name p2. Write a java code to run a main method and display the content of student result in appropriate format.

Package1:-

```
package p1;

public class stud
{
    public int rollno=26;
    public String name="Chetan puajri";
    public String clas ="BSCCS";
}
```

Package2:-

```
package p2;

import p1.stud;

class result extends stud
{
```

```

public static void main(String args[])
{
    stud s=new stud();
    int os=98;
    int java=95;
    int dbms=94;
    Int result =os+java+dbms;
    int per=result/3;
    System.out.println("result : "+result );
    System.out.println("-----
-----");
    System.out.println("RollNo \t Name \t\t class \t os \t dbms
\t java \t Total \t per%");
    System.out.println(s.rollno +"    "+s.name+" "+ s.clas+ "
"+os+"    "+dbms+"    "+java+"    "+result+"    "+per);
    System.out.println("-----
-----");
    System.out.println("");
}
}

```

Output :-

```

E:\sem_2\java\p1>java p2.result
result : 287
-----
RollNo    Name          class    os      dbms    java    Total    per%
26        Chetan puajri  BS CCS  98      94      95      287      95
-----

```

Ques.2

Create a package called “Salary”. Under this package create 2 classes say “income” and “expense”. The income class contains „salary, interest and rent“ while expense contains food, clothing and education“ expenses. Create a class “Budget” which uses these classes and count savings of a family. The Budget class is in “MyBudget” package.

Package:-1

```
package salary;  
  
public class income  
{  
    public int salary=10000;  
    public int interest=1000;  
    public int rent=20000;  
    public int totalincome=salary+interest+rent;  
}
```

Package:-2

```
package salary;  
  
public class expense  
{  
    public int food=9000;  
    public int cloth=3000;  
    public int edu=5000;  
    public int totalexpanse=food+edu+cloth;  
}
```

Main class:-

```
import salary.income;
import salary.expense;
public class mybudget
{
    public static void main(String args[])
    {
        income i=new income();
        expense e=new expense();
        int budget= i.totalincome - e.totalexpenditure;
        System.out.println("Salary \t interest\tRent \t Food
\tCloth \t Edu \t Budget");
        System.out.println( i.salary+" "+i.interest+"
+i.rent+" "+e.food+" "+e.cloth+" "+e.edu+ " "+budget);
        System.out.println("Monthly budget : "+budget);
    }
}
```

Output:-

```
E:\sem_2\java\p2\New folder>java mybudget
Salary    interest    Rent    Food    Cloth    Edu    budget
10000    1000    20000    9000    3000    5000    14000
Monthly budget : 14000

E:\sem_2\java\p2\New folder>_
```

Ques.3

Define a package called stores with class “item” which contains member variable itemno, name, quantity and cost. Create a class sale from item with member variable qty. Store inside a package purchase. Do the following :

=> Stock available

=> Total

=> View all

Package:- 1

```
package stores;

public class item
{
    public int itemno[]={101,112,123,134};
    public String name[]{"shirt","jeans","shoes","cap"};
    public int qut[]={10,10,10,9};
    public int cost[]={1000,2000,3000,500};
    public int stock=0;
    public void view()
    {
        for(int i=0;i<qut.length;i++)
        {
            stock=stock+qut[i];
        }
    }
}
```

```

        System.out.println("Stoack Available:" +stock);
        System.out.println("total :" +name.length);
        System.out.println("View all :");
        System.out.println("item_no \t item_name \t quantity \t
Cost");
        System.out.println("-----
-----");
        for(int i=0;i<name.length;i++)
        {
            System.out.println(itemno[i]+"      "+ name[i]+"
"+qut[i]+"      "+cost[i]);
        }
    }
}

```

Package:-2

```

package purchase;
import stores.item;
class sale extends item
{
    public static void main(String args[])
    {

```

```

        item i=new item();

        i.view();

    }

}

```

Output:-

```

E:\sem_2>javac -d . item.java
E:\sem_2>javac -d . sale.java
E:\sem_2>java purchase.sale
Stoack Available:39
total :4
View all :
item_no      item_name      quantity      Cost
-----
101          shirt          10            1000
112          jeans          10            2000
123          shoes          10            3000
134          cap            9             500
E:\sem_2>_

```

Ques.4

WAP to create simple thread by extending Thread class and by implementing Runnable interface.

```

class A implements Runnable
{
    public void run()
    {
        System.out.println("Hello Devil");
        System.out.println("Hello DEVII");
    }
}

class B implements Runnable
{

```

```
        public void run()
        {
            System.out.println("hello world 1");
            System.out.println("hello world 1");
        }
    }
class Q4
{
    public static void main(String[] args) {
        A t1 = new A();
        Thread a1 = new Thread(t1);
        B t2 = new B();
        Thread a2 = new Thread(t2);
        a1.start();
        a2.start();
    }
}
```

Output:-

```
E:\sem_2\java\p2\New folder>java Q4
Hello Devil
Hello DEVIl
hello world 1
hello world 1

E:\sem_2\java\p2\New folder>
```


Ques.5

Write a multi-threaded program which sets the priority the threads and gets the name the name of threads.

```
class A extends Thread
{
    String name;
    A(String name)
    {
        super(name);
    }
    public void run()
    {
        System.out.println("Hello Devil");
    }
}

class B extends Thread
{
    String name;
    B(String name)
    {
        super(name);
    }
    public void run()
    {
```

```
        System.out.println("hello world 2");
    }
}

public class q5 {
    public static void main(String[] args) {
        A t1 = new A("hello");
        B t2 = new B("Devil");
        t1.setPriority(Thread.MIN_PRIORITY);
        t2.setPriority(Thread.MAX_PRIORITY);
        t1.start();
        t2.start();

        System.out.println("Name of thread 1 is "+ t1.getName());
        System.out.println("Name of thread 2 is "+ t2.getName());
    }
}
```

Output:-

```
E:\sem_2\java\p2\New folder>java q5
Name of thread 1 is hello
hello world 2
Hello Devil
Name of thread 2 is Devil

E:\sem_2\java\p2\New folder>_
```

Ques.6 Write a multi-threaded program using sleep() and yield() methods.

```
class A extends Thread
```

```
{
```

```
    public void run()
```

```
    {
```

```
        for(int i =0;i<4;i++)
```

```
        {
```

```
            System.out.println("Hello Chetan");
```

```
        }
```

```
    }
```

```
}
```

```
class B extends Thread
```

```
{
```

```
    public void run()
```

```
    {
```

```
        for(int i =0;i<4;i++)
```

```
        {
```

```
            System.out.println("Good Morning");
```

```
            try
```

```
            {
```

```
                Thread.sleep(400);
```

```
            }
```

```
            catch (Exception e)
```

```

        {
            System.out.println(e);
        }
    }
}

class Q6
{
    public static void main(String[] args)
    {
        A t1 = new A();
        B t2 = new B();
        t1.start();
        t1.yield();
        t2.start();
    }
}

```

Output:-

```

E:\sem_2\java\p2\New folder>java Q6
Hello Chetan
Hello Chetan
Hello Chetan
Hello Chetan
Good Morning
Good Morning
Good Morning
Good Morning

```

Ques.7 Write a multi-threaded program using islive() and join() methods.

```
class Q7 extends Thread
{
    public void run()
    {
        Thread t = Thread.currentThread();
        System.out.println("Current thread: " + t.getName());
        if (t.isAlive() == true)
        {
            System.out.println("The Thread is Alive ");
        }
        else
        {
            System.out.println("The Thread is not Alive");
        }
    }
}

public static void main(String args[])
{
    Thread t = new Thread(new Q7());
    t.start();

    try
    {

```

```
        t.join(1000);
    }

    catch (Exception e)
    {
        System.out.println(e);
    }

    System.out.println("\nJoining after 1000" + " milliseconds: \n");

    System.out.println("Current thread: "+ t.getName());

    if (t.isAlive() == true)
    {
        System.out.println("the Thread is Alive ");
    }

    else

    {
        System.out.println("The Thread is not Alive");
    }

    }

}
```

Output:-

```
E:\sem_2\java\p2\New folder>java Q7
Current thread: Thread-1
The Thread is Alive

Joining after 1000 milliseconds:

Current thread: Thread-1
The Thread is not Alive
```

Ques.8

Write A Program that generates a custom exceptions if inputted marks is not valid.(marks should be between 1 to 100)

```
import java.lang.Exception;
import java.util.Scanner;
class MyException extends Exception
{
    MyException( String message)
    {
        super(message);
    }
}
public class Q8
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        int a;
        System.out.println("Enter your marks");
        a= sc.nextInt();
        try
        {
            if(a<1 || a>100)
```

```
        {
            throw new MyException("Invalid marks entered");
        }
        else
        {
            System.out.println("your marks is "+ a);
        }
    }
    catch(MyException e){
        System.out.println("exception");
        System.out.println(e.getMessage());
    }
}
}
```

Output:-

```
E:\sem_2\java\p2\New folder>java Q8
Enter your marks
98
your marks is 98

E:\sem_2\java\p2\New folder>java Q8
Enter your marks
101
exception
Invalid marks entered
```


Ques.9

Write A Program that generates a custom exception if any of its command line arguments are floating point number.

```
import java.lang.Exception;

public class Q9
{
    public static void main(String[] args)
    {
        try
        {
            float a= Float.parseFloat(args[0]);
        }
        catch(Exception e)
        {
            System.out.println("exception you entered invalid value");
        }
    }
}
```

Output:-

```
E:\sem_2\java\p2\New folder>java Q9 1

E:\sem_2\java\p2\New folder>java Q9 chetan
exception you entered invalid value
```

Ques.10

Write A Program that generates a custom exception if inputted age is not eligible for voting.

```
import java.lang.Exception;
import java.util.Scanner;

class MyException extends Exception
{
    MyException( String message)
    {
        super(message);
    }
}

public class Q10
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        int a;
        System.out.println("Enter your age");
        a= sc.nextInt();
        try
        {
            if(a<=18)
```

```

        {
            throw new MyException("Sorry you are not eligible for
vote or Invalid age entered");
        }

        else
        {
            System.out.println("your age is "+ a+"and You are eligible
for voting");
        }
    }
    catch(MyException e)
    {
        System.out.println("exception");
        System.out.println(e.getMessage());
    }
}
}

```

Output:-

```

E:\sem_2\java\p2\New folder>java Q10
Enter your age
15
exception
Sorry you are not eligible for vote or Invalid age entered

E:\sem_2\java\p2\New folder>java Q10
Enter your age
25
your age is 25and You are eligible for voting

```

Ques.11

Write a java program to simulate bank transactions. Take two variables, balanceAmount and withdrawAmount. Create a custom exception "InvalidTransaction", Your program must throw the "InvalidTransaction" exception if the withdrawAmount is less than balanceAmount.

```
import java.lang.Exception;

import java.util.Scanner;

class MyException extends Exception
{
    MyException( String message)
    {
        super(message);
    }
}

public class q11
{
    public static void main(String[] args)
    {
        Scanner s = new Scanner(System.in);
        int balanceAmount=5001,a;
        System.out.println("Enter your withdrawal amount");
        a= s.nextInt();
        try
        {
```

```

        if(balanceAmount <=a)
        {
            throw new MyException("Invalid Transaction");
        }
        else
        {
            System.out.println("Thanks for Withdrawan money
\n your curent balance is : "+(balanceAmount-a)+"rs");
        }
    }
    catch(MyException e)
    {
        System.out.println("exception");
        System.out.println(e.getMessage());
    }
}
}

```

Output:-

```

E:\sem_2\java\p2\New folder>java q11
Enter your withdrawal amount
5000
Thanks for Withdrawan money
your curent balance is : 1rs

E:\sem_2\java\p2\New folder>java q11
Enter your withdrawal amount
50002
exception
Invalid Transaction

```

Ques.12

Write A Program to copying character from one file into another.

```
import java.io.*;

class q12
{
    public static void main(String args[])
    {
        try
        {
            File in=new File("file.txt");
            File ot=new File("output.txt");
            FileReader ins=new FileReader(in);
            FileWriter ots=new FileWriter(ot);

            int ch;

            while((ch=ins.read())!=-1)
            {
                ots.write(ch);
            }

            ins.close();
            ots.close();
        }

        catch(Exception e)
        {
            System.out.println(e);
        }
    }
}
```

```
    }  
    System.out.println("Done!!!!!! the file copy sucessfully");  
}  
}
```

Output:-

```
E:\sem_2\java\p2\New folder>javac q12.java  
  
E:\sem_2\java\p2\New folder>java q12  
Done!!!!!! the file copy sucessfully  
  
E:\sem_2\java\p2\New folder>
```

Ques.13

Write A Program to writing bytes to a file.

```
import java.io.*;  
  
class Q13  
{  
  
    public static void main(String args[])  
    {  
  
        try  
        {  
  
            FileInputStream s=new FileInputStream("file.txt");  
  
            int i;  
  
            while((i=s.read())!=-1)
```

```

        {
            System.out.print((char)i);
        }

s.close();
}

        catch(Exception e)
        {
            System.out.println(e);
        }
    }
}

```

Output:-

```

E:\sem_2\java\p2\New folder>javac Q13.java

E:\sem_2\java\p2\New folder>java Q13
hello i'm devil!!!!
E:\sem_2\java\p2\New folder>_

```

Ques.14

_Write A Prpgram to reading bytes from a file.

```

import java.io.*;

class Q14
{
    public static void main(String args[])
    {

```



```
try
{
    FileInputStream s=new FileInputStream("file.txt");
    FileOutputStream o=new FileOutputStream("output.txt");
    int i;
    while((i=s.read())!=-1)
    {
        o.write((char)i);
    }
    s.close();
    o.close();
}
catch(Exception e)
{
    System.out.println(e);
}
System.out.println("DONE Byte program Run Sucessfully");
}
```

Output:-

```
E:\sem_2\java\p2\New folder>javac Q14.java

E:\sem_2\java\p2\New folder>java Q14
DONE Byte program Run Sucessfully
```

Ques.15

WAP to Copying bytes from one file to another.

```
import java.io.*;

class Q15
{
    public static void main(String args[])
    {
        try
        {
            FileInputStream in=new FileInputStream("file.txt");
            FileOutputStream ot=new FileOutputStream("output.txt");
            int ch;
            while((ch=in.read())!=-1)
            {
                ot.write(ch);
            }
            in.close();
            ot.close();
        }
        catch(Exception e)
        {
            System.out.println(e);
        }
    }
}
```

```
System.out.println("Done Copying Bytes sucessfully");  
}  
}
```

Output:-

```
E:\sem_2\java\p2\New folder>javac Q15.java  
  
E:\sem_2\java\p2\New folder>java Q15  
DONE Copying Bytes sucessfully
```