**Department of Information Science and Engineering**

**COURSE CODE: 19IS4PCJAV COURSE: JAVA PROGRAMMING**

**LAB PROGRAM LIST**

**AY – 2019-20**

1 a. Design and create a class named Retail Item that holds data about an item in a retail store. The class should have the following fields:

* Description - The description field references a String object that holds a brief description of the item.
* Units - The units field is an int variable that holds the number of units currently in inventory.
* Price - The price field is a double that holds the item’s retail price.

Write a constructor that accepts arguments for each field, appropriate mutator methods that store values in these fields, and accessor methods that return the values in these fields. Write the main method which creates three Retail Item objects and invokes appropriate methods.

import java.util.Scanner;

// import java.util.\*;

public class retail {

    Scanner sc= new Scanner(System.in);

    int unit;

    float price;

    String desc;

    retail(){

     unit = 1;

     price = 1;

    }

    void desc(){

            //System.in is a standard input stream

        System.out.println("Enter description of product :\n");

        desc = sc.next();

    }

    void input(){

        System.out.println("Enter Number of units :\n");

        unit = sc.nextInt();

        System.out.println("Enter price :\n");

        price = sc.nextFloat();

    }

    void disp(){

        System.out.println("Description of product : "+desc);

        System.out.println("Number of units : "+unit);

        System.out.println("total price : "+(unit\*price));

    }

}

class mainfun{

    public static void main(String[] args){

        retail r = new retail();

        r.disp();

        r.desc();

        r.input();

        r.disp();

    }

}

1b.Write a class named Car that has the following data members:

• **model**. The model field is an int that holds the car’s year.

• **make**. The make field references a String object that holds the make of the car.

• **speed**. The speed field is an int that holds the car’s current speed.

The class should have the following constructor and other methods.

• The constructor should accept the car’s year model, make and speed as arguments.

• Accessor methods should get the values stored in an object’s year, Model, make, and speed fields.

• Accelerate method should add 5 to the speed field each time it is called.

• Brake method should subtract 5 from the speed field each time it is called.

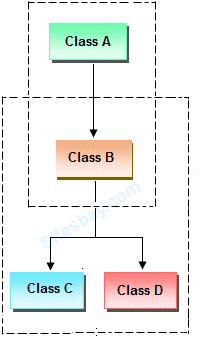
Demonstrate the class in a program that creates a Car object, and then calls the accelerate method five times. After each call to the accelerate method, get the current speed of the car and display it. Call the brake method five times. After each call to the brake method, get the current speed of the car and display it.

ANS:

class car  
{  
int model,speed;  
String make;  
  
car(int m,String s,int sp)  
{  
model = m;  
make = s;  
speed = sp;  
}  
  
void get\_values()  
{  
System.out.println("The model of the car is:"+model);  
System.out.println("The make of the car is:"+make);  
System.out.println("The speed of the car is:"+speed+"\n");  
}  
  
void accelerate()  
{  
speed += 5;  
System.out.println("The current speed of the car is:"+speed);  
}  
  
void brake()  
{  
speed -= 5;  
System.out.println("The current speed of the car is:"+speed);  
}  
  
}  
  
public class lab10  
{  
public static void main(String args[])  
{  
car c = new car(2018,"honda",20);  
c.get\_values();  
for(int i=0;i<5;i++)  
{c.accelerate();}  
for(int i=0;i<5;i++)  
{c.brake();}  
}  
}

2a. Identify the type of inheritance in the given diagram. Create a class A with two integer member variables that are private, two float variables that are protected and two integer variables that are public. Let class B inherit class A and class C and Class D are inherited from class B. Write appropriate methods to illustrate the following

1. Usage of super keyword
2. Function overriding
3. Default constructors
4. Parameterized constructors
5. How to we make a method not to be over ridden and a class not be inherited further

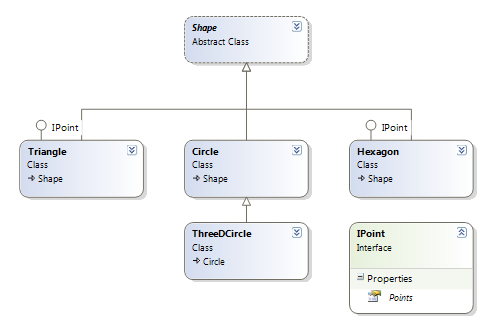


ANS

class A  
{  
private int a,b;  
protected float c,d;  
int e,f;  
  
A()  
{  
a = 1;  
b=2;  
c=3;  
d=6;  
e=4;  
f=5;  
}  
  
A(int x,int y,int z,int v,float g,float h)  
{  
a =x;  
b=y;  
c=g;  
d=h;  
e=z;  
f=v;  
}  
  
}  
  
class B extends A  
{  
int f;  
B()  
{  
f=10;  
}  
  
B(int k,int x,int y,int z,int v,float g,float h)  
{  
super(x,y,z,v,g,h);  
f = k;  
}  
  
void display()  
{  
System.out.println("this is base class f:"+super.f+" this is subclass f:"+f);  
}  
}  
  
final class C extends B  
{  
void display()  
{  
System.out.println("this is overriding function of class B in class C");  
}  
}  
  
class D extends B  
{  
final void display()  
{  
System.out.println("this is overriding function of class B in class D");  
}  
}  
  
public class lab15  
{  
public static void main(String args[])  
{  
B b = new B();  
b.display();  
  
B b1 = new B(1,2,3,4,5,6,7);  
b1.display();  
  
B c = new C();  
c.display();  
  
B d = new D();  
d.display();  
}}

2b. Define one class **A** in package **apack**. In class **A**, four variables are defined of access modifiers default, protected, private and public. Define class **B** in package **bpack** which extends **A** and write **display()** method which access variables of class **A**. Define class **C** in package **cpack** which has one method **display()** in that create one object of class **A** and display its variables. Define class **ProtectedDemo** in package **dpack** which contains the main () method. Create objects of class **B** and **C** and call display method for both these objects. Analyze the program by interpreting the access modifiers and provide valid conclusion.

2c. Design the given model in java



ANS

interface IPoint  
{  
void points(int x,int y);  
}  
  
abstract class shape1  
{  
abstract void type();  
}  
  
class circle extends shape1  
{  
void type()  
{  
System.out.println("the shape of this geometrical figure is circle");  
}  
}  
  
class threeDcircle extends circle  
{  
void type()  
{  
System.out.println("the shape of this geometrical figure is sphere");  
}  
  
void info()  
{  
System.out.println("a 3 dimensional circle is a sphere");  
}  
}  
  
class triangle extends shape1 implements IPoint   
{  
void type()  
{  
System.out.println("the shape of this geometrical figure is triangle");  
}  
  
public void points(int x,int y)  
{  
System.out.println("the co-ordinates("+x+","+y+") lies inside the triangle");  
}  
}  
  
class hexagon extends shape1 implements IPoint   
{  
void type()  
{  
System.out.println("the shape of this geometrical figure is hexagon");  
}  
  
public void points(int x,int y)  
{  
System.out.println("the co-ordinates("+x+","+y+") lies inside the hexagon");  
}  
}  
  
public class lab17  
{  
public static void main(String args[])  
{  
threeDcircle d = new threeDcircle();  
d.info();  
d.type();  
  
shape1 s = new circle();  
s.type();  
  
triangle t = new triangle();  
t.points(3,6);  
t.type();  
  
hexagon h = new hexagon();  
h.points(4,5);  
h.type();  
}  
}

Write a program to enter two numbers from the keyboard and pass the numbers to a method divide. The divide method, divides the first number with the second number and generates an arithmetic exception if the second number is 0.

3 a. Consider a student examination database system that prints the mark sheet of the students. Input the following from the command line student name and marks in 6 subjects. These marks should be in between 0 and 50 if the marks are not in the specified range raise a Range Exception else find the total marks and print the percentage of the student.

ANS

class RangeException extends Exception  
{  
String detail;  
RangeException(String a)  
{  
detail = a;  
}  
public String toString()  
{  
return detail;  
}  
}  
  
public class lab21  
{  
public static void main(String args[])  
{  
try{  
int marks=0;  
String name = args[0];  
  
for(int i=1;i<args.length;i++)  
{  
int k = Integer.parseInt(args[i]);  
if(k>50 || k<0){  
throw new RangeException("marks exceeded the limit");  
}  
else{marks+=k;}  
}  
System.out.println(marks);  
System.out.println("The percentage of "+name+" is "+((float)marks\*100/300));  
}  
catch(Exception e)  
{  
System.out.println(e);  
}  
}  
}

3b. Create a class temperature with member variable temp. Implement exception handling to test if temperature is equal to zero.

ANS

class temperature  
{  
int temp;  
  
temperature()  
{temp = 25;}  
  
temperature(int k)  
{temp=k;}  
}  
  
class TempException extends Exception  
{  
String detail;  
TempException(String a)  
{detail=a;}  
  
public String toString()  
{  
return detail;  
}  
}  
  
public class lab22  
{  
public static void main(String args[])  
{  
temperature t = new temperature(0);  
try  
{  
if(t.temp==0)  
{  
throw new TempException("temperature 0 exception raised");  
}  
else{  
System.out.println("the temperature is "+t.temp);  
}  
}  
catch(Exception e)  
{  
System.out.println(e);  
}  
}  
}

4 a. Consider a Bus reservation system that allows online reservations to its customers. Suppose there are two transactions of reservation for a particular seat simultaneously which leads to race condition. Develop a solution to avoid the unpredictable situation with a program.

ANS

import java.util.\*;  
  
class bus  
{  
 int seats[] = new int[10];  
  
public void booked(String owner,int seatNo)  
{  
try{  
if(seats[seatNo]==1)  
{  
System.out.println("sorry seat number "+seatNo+" has been already booked");  
}  
else{  
System.out.println("Seat number "+seatNo+" has been alloted to "+owner);  
seats[seatNo]=1;  
}  
}  
catch(ArrayIndexOutOfBoundsException e)  
{  
System.out.println(e);  
}  
}  
}  
  
class bookSeat extends Thread  
{  
String name;  
int num;  
bus b;  
  
bookSeat(String n,int number,bus b)  
{  
name = n;  
this.b = b;  
num = number;  
}  
  
public void run()  
{  
synchronized(b)  
{  
b.booked(name,num);  
}  
}  
}  
  
public class lab23  
{  
public static void main(String args[])  
{  
bus booking = new bus();  
bookSeat t1 = new bookSeat("mike",5,booking);  
bookSeat t2 = new bookSeat("jim",5,booking);  
//bookSeat t3 = new bookSeat("andy",5,booking);  
  
t1.start();  
t2.start();  
//t3.start();  
  
try{  
t1.join();  
t2.join();  
//t3.join();  
}  
catch(Exception e)  
{  
System.out.println("Inpterupted "+e);  
}  
}  
}

4 b. Create a class called Library. Write a program to manipulate the book information from files by using FileInputStream and FileOutputStream.

5 a. Write a program to implement dynamic growable queue using generics.

class queue<T>  
{  
T[] arr;  
int top;  
int size;  
  
queue(int s)  
{  
size = s;  
arr = (T[])new Object[size];  
top=-1;  
}  
  
void push(T var)  
{  
if(top==size-1){  
arr[++top] = var;  
}  
else  
{  
T[] a = (T[]) new Object[size\*2];  
  
for(int i=0;i<size;i++)  
{  
a[i] = arr[i];  
}  
this.size = size\*2;  
arr = a;  
arr[++top] = var;  
}  
  
}  
  
void print\_queue()  
{  
for(int i=0;i<top;i++)  
System.out.println(arr[i]);  
}  
}  
  
public class lab27  
{  
public static void main(String args[])  
{  
queue<Integer> iob = new queue<Integer>(5);  
  
for(int i=1;i<=20;i++)  
{  
iob.push(i);  
}  
  
iob.print\_queue();  
}  
}

5b.Write code to

1. chars arr={‘a’,’b’,’’c’.’d’,’e’,’f’}

String str= new String(arr,2,3);

String(String strObj)

1. str=”hello world”

str2=”bms college”

str=str1+str2;

1. override toString() function to print “good morning”
2. Assign a diiferent char at pos 2 in a string “hello”
3. Convert the string “ This is java programming” to array of characters
4. Given the strings

String s1= “hello world”

String s2=”hello world”

String s3=”bms college “

Write appropriate code to compare s1 with s2 and s1 with s3 for i)equal strings and also using compareTo function also compare given region of string s1 with string s2, demonstrate the difference between equals and ==.

1. To search a good in a string s1=”good morning” and s2=”morning is started”
2. Extract a substring good in the string s2=”good morning”
3. Replace Hello with greetings in the given string s1=”hello world”
4. Eliminate spaces in the string s1=”Hello BMSCE Good Morning”
5. Join the string s1={ “xyz”,24} with email id and phone number

ANS

class override{  
String a = "good";  
String b = "morning";  
@Override  
public String toString(){  
return a+" "+b;  
}  
}  
  
public class lab28{  
public static void main(String args[]){  
char[] arr={'a','b','c','d','e','f'};  
String str = new String(arr,2,3);  
System.out.println(str+"\n");  
  
String str1 = "hello world";  
String str2 = " bms college";  
System.out.println(str1.concat(str2)+"\n");  
  
override o = new override();  
System.out.println("overriding toString:"+o+"\n");  
  
StringBuilder str3 = new StringBuilder("hello");  
str3.setCharAt(2,'z');  
System.out.println(str3.toString()+"\n");  
  
String str4 = "This is java programming";  
arr = str4.toCharArray();  
System.out.println(arr);  
  
String str5 = "hello world";  
System.out.println("\nstr5.compareTo(str1):"+str5.compareTo(str1));  
System.out.println("str1.compareTo(str2):"+str1.compareTo(str2));  
System.out.println("str1==str5:"+(str1==str5));  
System.out.println("str1.equals(str5):"+str1.equals(str5)+"\n");  
  
String s1 = "good morning";  
String s2 = "morning is started";  
System.out.println(s1.indexOf("good"));  
System.out.println(s2.indexOf("good")+"\n");  
  
System.out.println(s1.substring(s1.indexOf("good"),"good".length())+"\n");  
  
str1 = "hello world";  
System.out.println(str1.replace("hello","greetings")+"\n");  
  
str2 = "Hello BMSCE good morning";  
System.out.println(str2.replace(" ","")+"\n");  
  
String[] arr1 = {"xyz","24"};  
String email = "abc@gmail.com";  
String number = "1234567890";  
System.out.println(email+arr1[0]);  
System.out.println(number+arr1[1]);  
}  
}

6a Develop a program to create a Array list and perform operation on it.

import java.util.\*;  
  
class lab29{  
public static void main(String args[]){  
ArrayList<Integer> arr = new ArrayList<Integer>();  
for(int i=1;i<=7;i++)  
arr.add(i);  
System.out.println(arr);  
arr.remove(4);  
System.out.println(arr);  
arr.add(1,5);  
System.out.println(arr);  
for(int i=0;i<arr.size();i++)  
System.out.print(arr.get(i)+" ");  
}  
}

6b Write a hash set for books with id, name, author, and quantity as attributes and perform operations on it.

import java.util.\*;  
  
class book{  
int id,quantity;  
String name,author,publisher;  
public book(int id,String name,String a,String p,int q){  
this.id=id;  
this.name=name;  
author=a;  
publisher=p;  
quantity=q;  
}  
}  
  
public class lab30{  
public static void main(String args[]){  
HashSet<book> arr = new HashSet<book>();  
book b1 = new book(1,"abc","xyz","pqr",5);  
book b2 = new book(2,"book2","author2","pub2",10);  
arr.add(b1);  
arr.add(b2);  
  
System.out.println("The book abc is present in the hashset: "+arr.contains(b1));  
  
for(book b:arr){  
System.out.println(b.id+" "+b.name+" "+b.author+" "+b.publisher+" "+b.quantity);  
}  
}  
}

6c Program to remove the highest and lowest value from a tree set

import java.util.\*;  
class lab31{  
public static void main(String args[])  
{  
 NavigableSet<Integer> num = new TreeSet<>();  
 num.add(5);  
 num.add(3);  
 num.add(1);  
 num.add(4);  
 num.add(2);  
 System.out.println("initial treeset: "+num);  
 System.out.println("the lowest element is "+num.pollFirst());  
 System.out.println("After removing the lowest element: "+num);  
 System.out.println("the highest element is "+num.pollLast());  
 System.out.println("After removing the highest element: "+num);   
}  
}

6d Program to demonstrate add, add first, addlast, clear and display the elements in ArrayDeque

import java.util.\*;  
class lab32{  
public static void main(String args[]){  
 ArrayDeque<String> ar = new ArrayDeque<String>();  
 ar.add("name1");  
 ar.add("name2");  
 ar.add("name3");  
   
 ar.addFirst("name4");  
 ar.addFirst("name5");  
   
 ar.addLast("name6");  
 ar.addLast("name7");  
   
 for(Iterator itr = ar.iterator();itr.hasNext();)  
 System.out.print(itr.next()+" ");  
 System.out.println("");   
 ar.clear();  
 System.out.println("After clearing:");  
 for(Iterator itr=ar.iterator();itr.hasNext();)  
 System.out.println(itr.next());  
  
}  
}

7a.Develop a registration form using swings/simple calculator.

7b.Develop a program in java using swing to obtain a similar form output that is given below

