*JAVA ASSIGMENT = 2*

1. Define class Human with first name and last name. Define  
new class Student which is derived from Human and has new field – grade. Define  
class Worker derived from Human with new field weekSalary and work-hours per  
day and method MoneyPerHour() that returns money earned by hour by the worker.  
Define the proper constructors and properties for this hierarchy. Create object  
and demonstrate.

class human

{

String firstname,lastname;

human(String fn,String ln)

{

firstname=fn;

lastname=ln;

}

}

class student extends human

{

String grade;

student(String fn,String ln,String g)

{

super(fn,ln);

grade=g;

}

void display\_student()

{

System.out.println("student firstname :="+firstname);

System.out.println("student lastname :="+lastname);

System.out.println("Grade := "+grade);

}

}

class worker extends human

{

int salary,hours;

worker(String fn,String ln,int s,int h)

{

super(fn,ln);

salary=s;

hours=h;

}

double MoneyPerHour()

{

return ((double)salary/(hours\*7));

}

void display\_worker()

{

System.out.println("worker firstname :="+firstname);

System.out.println("worker lastname :="+lastname);

System.out.println("worker salary :="+salary);

}

}

class humandemo

{

public static void main(String a[])

{

student s=new student("parth","patel","A+");

s.display\_student();

worker w=new worker("satish","koringa",27000,7);

w.display\_worker();

//System.out.println(""+display\_student());

System.out.println("Per day salary := "+w.MoneyPerHour());

}

}

(2.)WAP to perform mathematical operation create a class  
called AddSub with attributes num1 and num2 of type int, also define methods  
calAdd( ) and calSub( ) to calculate addition and subtraction .Create another  
class MulDiv that extends AddSub class to use member a data of super class.  
MulDiv should have methods calMul( ) & calDiv( ) to calculate  
multiplication and division. A main method should access the methods and  
perform operations.

class AssSub

{

int num1,num;

void calAdd(int n,int n1)

{

int add;

num=n;

num1=n1;

add=num+num1;

System.out.println("Addition is : "+add);

}

void calSub(int n2,int n3)

{

int sub;

num=n2;

num1=n3;

sub=num-num1;

System.out.println("Subtraction is :"+sub);

}

}

class MulDiv extends AssSub

{

double value,value1;

void calMul(double v,double v1)

{

value=v;

value1=v1;

}

void calDiv(double v2,double v3)

{

value=v2;

value1=v3;

}

void display()

{

System.out.println("Divison:"+value\*value1);

System.out.println("Multiplication:"+value/value1);

}

}

class mathematical\_operation

{

public static void main(String a[])

{

// AssSub n=new AssSub();

MulDiv m=new MulDiv();

m.calAdd(20,20);

m.calSub(30,20);

m.calDiv(10,2);

m.calMul(1,2);

m.display();

}

}

(3.)Create a Shape class as the abstract class with abstract  
method draw( ), its implementation is provided by the Rectangle & Circle  
classes. Create a reference of Shape class and if you create the instance of  
Rectangle class, draw() method of Rectangle class will be invoked. And same for  
Circle class. (Dynamic Method Dispatch)

abstract class shape

{

abstract void drew();

}

class rectangle extends shape

{

void drew()

{

System.out.println("method of rectangle..");

}

}

class circle extends shape

{

void drew()

{

System.out.println("Dynemic Method Dispatch.");

}

}

class program3

{

public static void main(String a[])

{

shape s;

s=new rectangle();

s.drew();

s=new circle();

s.drew();

}

}

(4.)Create two interfaces Printable and Showable having  
methods print( ) and show( ) respectively. Create a Test\_Multiple class which  
implements all 2 interfaces and override print( ) and show( ) methods. Write a  
main method in this class and create an object of this class and use the  
methods.

interface printtable

{

void print();

}

interface showble extends printtable

{

void show();

}

class test\_method implements showble

{

public void print()

{

System.out.println("first interface print..");

}

public void show()

{

System.out.println("secound interface print...");

}

}

class program4

{

public static void main(String a[])

{

test\_method p=new test\_method();

p.print();

p.show();

}

}

(5.)Write a program that finds the length of the string “Java  
Programming”. And also display a sub-string formed by the last five characters  
of the string. ( use String class)

class shape

{

public static void main(String a[])

{

String str=new String("java program");

System.out.println(String="+str);

int i=str.length();

System.out.println("length="+i);

String s=str.substring(1,3);

System.out.println("substring="+s);

}

}

(6.) Create a package MathPack having class MathDemo with method add() and sub() to find addition andsubtraction. Create another program and `import package and invoke methods.

import MathPack.\*;

class program6

{

public static void main(String a1[])

{

methdemo m=new methdemo();

//m.add(10,20);

System.out.println(""+m.add(10,20));

//m.sub(20,10);

System.out.println(""+m.sub(20,10));

}

}

…..

package MathPack;

public class methdemo

{

int a,b;

int x,y;

public int add(int a,int b)

{

return a+b;

}

public int sub(int x,int y)

{

return x-y;

}

}

7. Write a program to do the following using in-built  
methods in the string class of Java. a. Find the 3rd character in the string  
"University" b. Find the index of character ‘o’ in string "Java  
Programming". c. Convert the string "Be Positive" to uppercase.  
d. Replace character 'l' with 'i' in the string "hello”.

class Program7

{

public static void main(String a[])

{

String s1=new String();

s1="University";

char c=s1.charAt(2);

System.out.println(c);

s1="java program";

int x=s1.indexOf("o");

System.out.println(x);

s1="be positive";

String s2=s1.toUpperCase();

System.out.println(s2);

s1="hello";

String s3=s1.replace('l','i');

System.out.println(s3);

}

}

9. Declare a class called book having author\_name as private  
data member. Extend book class to have two sub classes called book\_publication  
& paper\_publication. Each of these classes have private member called  
title. Write a complete program to show usage of dynamic method dispatch (dynamic  
polymorphism) to display book or paper publications of given author. Use  
command line arguments for inputting data.

//import java.util.\*;

class book

{

private String Auther\_Name;

book(String n)

{

Auther\_Name=n;

//Scanner sc=new Scanner(System.in);

//System.out.println("Enter Name:-"+Auther\_Name);

//s=sc.next();

}

void print()

{

System.out.println("Auther\_Name:-"+Auther\_Name);

}

}

class book\_publication extends book

{

private String title;

book\_publication(String n,String t)

{

super(n);

title=t;

}

void print()

{

System.out.println("titlt:-"+title);

}

}

class paper\_publication extends book

{

private String title1;

paper\_publication(String n,String t)

{

super(n);

title1=t;

}

void print()

{

System.out.println("title1:-"+title1);

}

}

class program9

{

public static void main(String args[])

{

//book b=new book()

book b1=new book\_publication(args[0],args[1]);

b1.print();

b1=new paper\_publication(args[2],args[3]);

b1.print();

}

}

10. The abstract Vegetable class has three subclasses named  
Potato, Brinjal and Tomato. Write an application that demonstrates how to  
establish this class hierarchy. Declare one instance variable of type String  
that indicates the color of a vegetable. Create and display instances of these  
objects. Override the toString() method of Object to return a string with the  
name of the vegetable and its color.

abstract class Vegetable

{

String color;

Vegetable(String color)

{

this.color=color;

}

public abstract String toString();

}

class Potato extends Vegetable

{

Potato(String color)

{

super(color);

}

public String toString()

{

return color;

}

}

class Brinjal extends Vegetable

{

Brinjal(String color)

{

super(color);

}

public String toString()

{

return color;

}

}

class Tomato extends Vegetable

{

Tomato(String color)

{

super(color);

}

public String toString()

{

return color;

}

}

class program1o

{

public static void main(String arge[])

{

Vegetable v1=new Potato("yellow");

System.out.println("Poatato color="+v1.toString());

v1=new Brinjal("dark purplish red");

System.out.println("Brinjal color="+v1.toString());

v1=new Tomato("red");

System.out.println("Tomato color="+v1.toString());

}

}

11. Create first class with following specifications: Class  
Name: Student Data Members: Enrollment\_No, Student\_Name, Semester Member  
Functions: GetStudentDetails () Create second class with following  
specifications: Class Name: Result Data Members: Enrollment\_No, CPI and SPI  
Member Functions: GetResultDetails () and DisplayResult () DisplayResult method  
header must be: public void DisplayResult (Student s).

import java.util.\*;

class Student

{

long Enrollment;

String Student\_name;

int Semester;

void GetStudent\_details()

{

Scanner sc=new Scanner(System.in);

System.out.print("Enter Enrollment\_no:-");

Enrollment=sc.nextInt();

System.out.print("Enter Name:-");

Student\_name=sc.next();

System.out.print("Enter Semester:-");

Semester=sc.nextInt();

}

}

class Result

{

//long Enrollment;

double SPI,CPI;

void GetResultDetails()

{

Scanner sc=new Scanner(System.in);

//System.out.print("Enter Enrollment\_no:-");

//Enrollment=sc.nextInt();

System.out.print("Enter SPI:-");

SPI=sc.nextDouble();

System.out.print("Enter CPI:-");

CPI=sc.nextDouble();

}

public void DisplayResult(Student s)

{

System.out.println("Enter Enrollment\_no:-"+s.Enrollment);

System.out.println("Enter Name:-"+s.Student\_name);

System.out.println("Enter Semester:-"+s.Semester);

System.out.println("Enter SPI:-"+SPI);

System.out.println("Enter CPI:-"+CPI);

}

}

class prodram11

{

public static void main(String args[])

{

Student s=new Student();

s.GetStudent\_details();

Result r=new Result();

r.GetResultDetails();

r.DisplayResult(s);

}

}

12. Create a class named 'Member' having the following  
members: 1 - Name 2 - Age 3 - Phone number 4 - Address 5 -    Salary It also has a  
method named 'printSalary' which prints the salary of the members. Two classes  
'Employee' and 'Manager' inherits the 'Member' class. The 'Employee' and  
'Manager' classes have data members 'specialization' and 'department'  
respectively. Now, assign name, age, phone number, address and salary to an  
employee and a manager by making an object of both of these classes and print  
the same along with specialization and department respectively.

class Member

{

String Address,Name;

long Phone\_Number;

int Salary,Age;

Member(String A,String N,long P,int Ag,int Sal)

{

Address=A;

Name=N;

Phone\_Number=P;

Age=Ag;

Salary=Sal;

}

void printSalary()

{

System.out.println("Entert Salary:-"+Salary);

}

}

class Employee extends Member

{

String Specialization;

Employee(String A,String N,long P,int Ag,int Sal,String Specialization)

{

super(A,N, P, Ag, Sal);

this.Specialization=Specialization;

}

void Print\_Employee()

{

System.out.println("\nEntert Address:-"+Address);

System.out.println("Entert Name:-"+Name);

System.out.println("Entert Phone\_Number:-"+Phone\_Number);

System.out.println("Entert Age:-"+Age);

System.out.println("Entert Specialization:-"+Specialization);

}

}

class Manager extends Member

{

String Department;

Manager(String A,String N,long P,int Ag,int Sal,String Department)

{

super( A, N,P, Ag, Sal);

this.Department=Department;

}

void Print\_Manager()

{

System.out.println("\nEntert Address:-"+Address);

System.out.println("Entert Name:-"+Name);

System.out.println("Entert Phone\_Number:-"+Phone\_Number);

System.out.println("Entert Age:-"+Age);

System.out.println("Enter Department:-"+Department);

}

}

class program12

{

public static void main(String args[])

{

Manager M=new Manager("Rajkot","Satish",123456,20,20000,"Programing");

Employee E=new Employee("Surat","Parth",765432,20,30000,"Bca");

M.Print\_Manager();

E.Print\_Employee();

}

}

13. WAP to find length of a string without using built-in  
function.

import java.util.\*;

class program13

{

public static void main(String a[])

{

Scanner sc=new Scanner(System.in);

System.out.print("enter string :=");

String s=sc.next();

// System.out.println("length="+s.length());

int length=0;

for(char c:s.toCharArray())

{

//System.out.println(length);

length++;

}

System.out.println("String length is ="+length);

}

}

14. WAP that checks whether a given string is a palindrome  
or not.

import java.util.\*;

class pelindrom

{

public static void main(String args[])

{

String str,rev="";

Scanner sc=new Scanner(System.in);

System.out.println("Enter String:-");

str=sc.next();

int chr=str.length();

for(int i=chr-1;i>=0;i--)

{

rev=rev+str.charAt(i);

}

System.out.println(rev);

if(str.equals(rev))

{

System.out.print("this pelindrom .."+str);

}

else

{

System.out.print("this not pelindrom .."+str);

}

}

}