

Output :

Model : Santro Xing

Year of Manufacturing : Santro Xing

Color : Red

Top Speed : 100

Qty Sold : 500

Model : Alto

Year of Manufacturing : Alto

Color : White

Top Speed : 98

Qty Sold : 750

Model : Wagon R

Year of Manufacturing : Wagon R

Color : Red

Top Speed : 98

Qty Sold : 400

Experiment:

Practical - 1

Date _____
Page No. 1

Consider we have a class of Cars under which Santro Xing, Alto and Wagon R represents individual objects. In this context each car object will have its own, Model, Year of Manufacture, Colour, Top Speed, etc. which form properties of the Car class and the associated actions i.e., object functions like Create(), Sold(), Display() form the Methods of Car Class. Use this class to create another class Company that tracks the models it creates.

Sol :

Class Car

{

String model;

int yearofmfg;

String color;

int topspeed;

int qtysold;

void Create (String m1, int y1, String c1, int sp)

{

model = m1;

yearofmfg = y1;

color = c1;

topspeed = sp;

}

CLASSTIME

Experiment :

Date _____
Page No. 2

Void Sold (int s1)

{

Qtysold = s1;

}

void Display ()

{

System.out.println ("Model:" + Model);

System.out.println ("Year of Manufacturing:" + model);

System.out.println ("Color:" + color);

System.out.println ("Top Speed:" + topspeed);

System.out.println ("Qty Sold:" + qtysold);

}

}

public class Company {

public static void main (String [] args)

{

// TODO Auto-generated method stub

Car santro = new Car();

Car alto = new Car();

Car wagonr = new Car();

santro.Create ("Santro Xing", 2019, "Red", 100);

santro.Sold (500);

santro.Display ();

alto.Create ("Alto", 2018, "White", 98);

alto.Sold (750);

alto.Display ();

wagonr.Create ("Wagon R", 2017, "Red", 98);

Experiment :

Date _____

Page No. 3

wagons. Sold (400);

wagons. Display (1);

?

?

Output:

Employee Details

Name of Employee : Rohit

Id of Employee : 1

Works of Employee : Software Testing

Perks of Employee : 50000

Role of Employee : Software Engineers

Responsibility of Employee : Software Testing & Maintenance

Employee Details

Name of Employee : Rajesh

Id of Employee : 2

Works of Employee : Software Development

Perks of Employee : 75000

Role of Employee : Sr. Software Engineers

Responsibility of Employee : Software Designing

Experiment:

Practical - 2

Date _____
Page No. 4

In a software company Software Engineers, Sr. Software Engineers, Module Lead, Technical lead, Project Lead, Project Manager, Program Manager, Directors all are the employees of the company but their work, perks, roles, responsibilities differs. Create the Employee base class would provide the common behaviour of all types of employee and also some behaviour properties that all employee must have for that company. Also include search method to search an employee by name.

Sol.

```
import java.util.*;  
class Employee  
{  
    String name;  
    int id;  
}  
class Typeemployee extends Employee  
{  
    String work;  
    int perks;  
    String roles;  
    String resp;  
    void setData(String sname, int id, String swork,  
    CLASSTIME
```

Employee Details

Name of Employee: Harsh

Id of Employee: 3

Works of Employee: Technical Advisor

Perks of Employee: 60000

Role of Employee: Module Lead

Responsibility of Employee: Supports and operations

Employee Details

Name of Employee: Vinod

Id of Employee: 4

Works of Employee: Technical Asst.

Perks of Employee: 60000

Role of Employee: Technical Lead

Responsibility of Employee: Supports & Operations

Employee Details

Name of Employee: Amarjeet

Id of Employee: 5

Works of Employee: Project Planning

Perks of Employee: 78000

Role of Employee: Project Lead

Responsibility of Employee: Project Supervision

Experiment:

Date _____
Page No. 5

```
intperks, String roles, String resp);
```

```
{
```

```
name = name;
```

```
id = sid;
```

```
work = swork;
```

```
perks = sp perks;
```

```
roles = sroles;
```

```
resp = sresp;
```

```
}
```

```
void displayDetails()
```

```
{
```

```
System.out.println("Name of Employee:" +  
name);
```

```
System.out.println("Id of Employee:" + id);  
System.out.println("Works of Employee:" +  
work);
```

```
System.out.println("Perks of Employee:" +  
perks);
```

```
System.out.println("Role of Employee:" +  
roles);
```

```
System.out.println("Responsibility of  
Employee:" + resp);
```

```
String getName()
```

```
{
```

```
return name;
```

```
}
```

```
}
```

CLASSTIME

Experiment :

Date _____
Page No. 6

```
Public class SoftwareCompany {  
    Public static void main (String ar [ ])  
    {  
        Type Employee emp [ ] = new TypeEmployee [ 8 ];  
        int i, j = 0;  
        String name;  
        for (i = 0; i < 8; i++)  
        {  
            Emp [ i ] = new TypeEmployee ();  
            emp [ 0 ]. setData ("Rohit", 1, "Software  
Testing", 50000, "Software Engineers", "Software  
Testing & Maintenance");  
            emp [ 1 ]. setData ("Rajesh", 2, "Software  
Developer", 75000, "Software Engineers",  
"Software Designing");  
            emp [ 2 ]. setData ("Harish", 3, "Technical  
Advisor", 60000, "Module Lead", "Support  
& operations");  
            emp [ 3 ]. setDate ("Vinod", 4, "Technical  
Asst", 60000, "Technical Lead", "Support &  
operations");  
            emp [ 4 ]. setData ("Amarjeet", 5, "Project  
Planning", 78000, "Project Lead", "Project  
Supervision");  
            emp [ 5 ]. setData ("Chandan", 6, "Project  
Advisor", 70000, "Project Manager", "Project
```

Experiment :

Date _____
Page No. 7

Development");

{ for (i=0; i<8; i++)

System.out.println ("Employee Details");

emp[i].displayDetails();

System.out.println ("-----");

}

System.out.println ("Enter name of employee
to search");

Scanner sc = new Scanner (System.in);

sname = sc.next();

for (i=0; i<8; i++)

{ if (emp[i].getName() == sname)

}

System.out.println ("Employee is searched");

emp[i].displayDetails();

f = 1;

}

}

If (f==0) {

System.out.println ("Employee is not found");

}

}

}

Output:

er

address:

jal

arrival time:

14:8

the records of arriving planes are
name is er address is jal id is 101 and arrival
time is 14:8

the counter is recorded with 1

enter the number of records do you want to enter
1

enter the name, id, address and departure time
of planes

name:

rer

id:

114

address:

ams

Suppose the Airport personals want to
maintain records for the arrival and depart-
ure of the planes. Create a class Airport
that has data like name, id, and address.

Create two more classes for Arrival and
Departure implementing Airport that will have
track of planes (their name, id, arrival time
or departure time and a counter to count
the number of arrivals) also include the
necessary methods to access these informations.

Also try to keep record of passengers to by
creating a new class Passenger. Also include a
method search() in Airport class to search any
passenger by name.

Sol.

```
package javaapplication25;  
import java.util.Scanner;
```

```
interface Airport  
{
```

```
    void read();
```

```
    void show();
```

```
}
```

```
class Arrival implements Airport  
{
```

departure time:

17:5

the records of departure plane are
name is rer address is ams id is 114 and
arrival time is 17:5

the counter is recorded with 1
enter the number of passengers

1

enter the passenger details

name:

rer

address:

dfd

id:

14

arriving time:

15:14

departure time:

16:78

destination:

delhi

the details of the passengers are

the name is rer id is 14 the address ddf
arriving time is 15:14 departure time is
16:78 and destination is delhi

enter the id of the passenger do you

Experiment:

Date _____
Page No. 9

{

Scanner sc = new Scanner (System. in);

int n = sc.nextInt();

int counter = 0;

String name [] = new String [n];

String address [] = new String [n];

int id [] = new int [n];

String arrtime [] = new String [n];

② Override

public void read () {

System.out.println ("enter the id, name,
address and arrival time of the plane");

for (int i = 0; i < n; i++)

{

System.out.println (" Id: ");

id [i] = sc.nextInt();

System.out.println ("name");

name [i] = sc.next();

System.out.println ("address");

address [i] = sc.next();

System.out.println ("arrival time");

arrtime [i] = sc.next();

counter ++;

}

}

② Override

public void show () {

CLASSTIME

want to search

14

the passenger is found

the name is ren address is dfol arriving
time 15:14 the departure at 18:78 and
destination is delhi

BUILD SUCCESSFUL (total time: 1 minute
10 seconds)

Experiment:

Date _____

Page No. 10

```
System.out.println ("The records of arriving  
planes are ");
```

```
for (int i=0; i<n; i++)  
{}
```

```
System.out.println ("name is " + name[i] +  
"address is " + address[i] + "id is " + id[i]  
+ "and arrival time is " + arritime[i]);
```

```
} System.out.println ("The counter is  
recorded with " + counter);
```

```
}
```

```
}
```

Class Department implements Airport

```
{ int counter = 0;
```

```
Scanner sc = new Scanner (System.in);  
int n = sc.nextInt();
```

```
String name [] = new String [n];
```

```
String address [] = new String [n];
```

```
int id [] = new int [n];
```

```
String deptime [] = new String [n];
```

@Override

```
public void read () {
```

```
System.out.println ("enter the name, id,  
address and departure time of place");
```

```
for (int i=0; i<n; i++)  
{}
```

```
System.out.println ("name");  
name [i] = sc.next();
```

CLASSTIME

```
System.out.println("address");  
address[i] = sc.nextInt();
```

```
System.out.println("departure time");  
deptime[i] = sc.nextInt();  
counter = counter + 1;  
}
```

{

@ Override

```
public void show() {
```

```
System.out.println("The records of departure plane are");
```

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

```
System.out.println("name is " + name[i] +  
"address is " + address[i] + "id is " + id[i]  
+ "and arrival time is " + deptime[i]);
```

```
System.out.println("the counter is  
recorded with " + counter);
```

{

{

```
class Passenger implements Airport
```

{

```
int a, d;
```

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

```
int n = sc.nextInt();
```

```
String name[] = new String[n];
```

```

String address [] = new String [n];
int planeid [] = new int [n];
String arrivingtime [] = new String [n];
String deptime [] = new String [n];
String destination [] = new String [n];

```

(a) Override

```

public void read () {
    System.out.println ("enter the passenger
details ");
    for (int i = 0; i < n; i++) {
        System.out.println ("name:");
        name [i] = sc.next ();
        System.out.println ("address");
        address [i] = sc.next ();
        System.out.println ("id:");
        planeid [i] = sc.nextInt ();
        System.out.println ("arriving time:");
        deptime [i] = sc.next ();
        System.out.println ("destination:");
        destination [i] = sc.next ();
    }
}

```

(b) Override

```

public void show () {
    System.out.println ("the details of the
passengers are ");
}

```

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

```
{  
    System.out.println ("the name is " + name[i]  
    + " id is " + planeid[i] + " address " + address[i] +  
    " arriving time [i] + " departure time is " +  
    depTime[i] + " and destination is " +  
    destination[i]);  
}
```

```
public void search()  
{
```

```
System.out.println ("enter the id of the  
passenger do you want to search ");
```

```
a = sc.nextInt();
```

```
for (int i=0; i<n; i++)  
{
```

```
if (planeid[i] == a)  
{
```

```
System.out.println ("the passenger is found");
```

```
d = i;
```

```
}
```

```
else
```

```
{
```

```
System.out.println ("passenger not found");
```

```
} }
```

```
{
```

```
public class JavaApplication25 {  
    public static void main (String [] args) {  
        System.out.println ("enter the number of  
details do you want to enter");  
        Arrival ar = new Arrival ();  
        ar.read ();  
        ar.show ();  
        System.out.println ("enter the number of  
passengers");  
        Passenger ps = new Passenger ();  
        ps.read ();  
        ps.show ();  
        ps.search ();  
    }  
}
```

Output:

name :

sham

address :

jal

destination :

clerk

salary :

10000

record of patients

registration id is : 121 age : 54 name : sham
address : jal disease : cough

the records you entered for staff

the registration id : 121 name : sham address : jal

designation : clerk salary : 10000

BUILD SUCCESSFUL (total time: 40 seconds)

Experiment :

Practical - 4

Date _____
Page No. 15

Create a whole menu driven hospital management system using concept of OOP like classes, inheritance. Include information about the following :

(i) Patient - name, registration id, age, disease, etc.

(ii) Staff - id, name, designation, salary, etc.

Sol.

```
package javaapplication26;  
import java.util.Scanner;  
class Patient  
{  
    Scanner sc = new Scanner (System.in);  
    int n = sc.nextInt();  
    int id [] = new int [n];  
    int age [] = new int [n];  
    String name [] = new String [n];  
    String address [] = new String [n];  
    String disease [] = new String [n];  
    void read ()
```

```
    {  
        System.out.println ("enter the details of the  
        patient ");  
    }
```

```
    System.out.println ("enter the registration id,  
    age, name, address and disease ");  
}
```

Experiment:

Date _____
Page No. 16

```
{ for(int i=0; i<n; i++)
```

```
    System.out.println("registration id:");  
    id[i] = sc.nextInt();
```

```
    System.out.println("age:");  
    age[i] = sc.nextInt();
```

```
    System.out.println("name:");  
    name[i] = sc.next();
```

```
    System.out.println("address:");  
    address[i] = sc.next();
```

```
    System.out.println("disease:");  
    des[i] = sc.next();
```

```
    System.out.println("salary:");  
    salary[i] = sc.nextInt();
```

```
}
```

```
}
```

```
void show1()
```

```
{
```

```
    super.show()
```

```
{
```

```
    super.show();
```

```
    System.out.println("the records you  
entered for staff");
```

```
{ for (int i=0; i<n; i++)
```

```
    System.out.println("the registration id:" + id[i] +  
    "name:" + name[i] + "address:" + address[i] +
```

Experiment :

Date _____

Page No. 17

"designation:" + des[i] + "salary:" + salary[i]);
}

}

}

```
public class JavaApplication26 {  
    public static void main (String [] args) {  
        System.out.println ("enter the number of  
records do you want to enter");  
        staff st = new staff ();  
        st.read1 ();  
        st.show1 ();  
    }  
}
```

Output:

run:

```
the instruments in string    veena  
the instruments in string    guitar  
the instruments in string    sitar  
the instruments in string    sarod  
the instruments in string    mandolin  
the instruments in wind     flute  
the instruments in wind     clarinet  
the instruments in wind     saxophone  
the instruments in wind     nadhaswaram  
the instruments in wind     piccolo  
the instruments in perc      tabla  
the instruments in perc      mridangam  
the instruments in perc      bangos  
the instruments in perc      drums  
the instruments in perc      tambour  
relevant details of the instrument  
BUILD SUCCESSFUL (total time: 1 second)
```

Experiment:

Practical - 5

Date _____
Page No. 18

Create a class called Musicians to contain three methods during string(), wind() and perc(). Each of three methods should contain initialize a string array to contain the following instruments:

- veena, guitar, sitar, sarod and mandolin under string()
- flute, clarinet saxophone, nadhaswaram and piccolo under wind()
- tabla, mridangam, bangos, drums and fambour under perc()

It should also display the contents of the arrays that are initialized. Create a derived class called TypeInsto contain a method called get() and show(). The get() method must display a means as follows:

Types of instruments to be displayed :

- i) String Instruments
- ii) Wind Instruments
- iii) Percussion Instruments

The show() method should display the relevant detail according to our choice. The base class variables are must be accessible only to its derived classes.

Sol.

package javaapplication24;

CLASSTIME

class Musicians

{

String arr[];

String brr[];

String crr[];

protected void string()

{

arr = new String [] {"veena", "guitar", "Sitar", "Ocarid",
"mandolin"};

}

protected void wind()

{

brr = new String [] {"flute", "clarinet", "saxo
phone", "nadaswaram", "piccolo"};

}

protected void perc()

super.string();

for (int i=0 ; i<arr.length ; i++)

{ System.out.println ("the instructions in
string "+ arr[i]); }

super.wind();

for (int i=0 ; i<brr.length ; i++)

System.out.println ("the instruments in wind"+
brr[i]);

}

Super.perc();

Experiment :

Date _____
Page No. 20

```
for (int i=0; i<arr.length; i++)  
{ System.out.println ("the instrument is perc " + arr[i]);  
}  
  
}  
void show()  
{  
System.out.println ("relevant details of the  
Instrument");  
}  
  
}  
  
}  
}  
public class JavaApplication24 {  
public static void main(String[] args) {  
  
TypoInstoy = new TypoInsto();  
ty.get(); ty.show();  
}  
}
```

Output:

run:
enter the number of records do you want to
enter for students
3
enter the name and age of the person

name:
rt

age:
45

name:
rt

age:
45

name:
ty

age:
45

enter the roll number and registration
number of the students

roll number:

45

Write three derived classes inheriting functionality of base class person (should have a member function that ask to enter name and age) and with added unique features of student , and employee , and functionality to assign , change and delete records of student and employee .

Sol.

```
package javaapplication27;  
import java.util.Scanner;  
import java.util.Arrays;
```

```
class person
```

```
{
```

```
Scanner sc = new Scanner (System.in);  
int n = sc.nextInt();
```

```
String name [] = new String [n];  
int age [] = new int [n];
```

```
int roll [] = new int [n];
```

```
int reg [] = new int [n];
```

```
int salary [] = new int [n];
```

```
String prof [] = new String [n];
```

```
int a , count = 0;
```

```
String me;
```

```
void read ()
```

registration number:

12

roll number:

45

registration number:

78

the roll number : 45 registration no is 78

the roll number : 45 registration no is 12

the roll number : 45 registration no is 78

enter the name of the person do you want
to delete

rt

record found

the records after deleting : name : rt age : 45

roll number : 45 registration number : 12

the records after deleting : name : ty age : 45

roll number : 45 registration number : 78

enter the name of the person do you want
to enter for employee

3

enter the name and age of the person

name :

gh

age :

45

name :

fdg

age :

Experiment:

Date _____

Page No. 22

{

System.out.println("enter the name and age of
the person");

for (int i=0; i<n; i++)

{

System.out.println("name:");

name[i] = sc.next();

System.out.println("age?");

age[i] = sc.nextInt();

}

3

void show()

{

for (int i=0; i<n; i++)

{

System.out.println("the name is" + name[i] +
"and age is" + age[i]);

}

}

void change()

{

3

void delete()

{ System.out.println("enter the name of the
person do you want to delete");

ne = sc.next();

for (int i=0; i<n; i++)

CLASSTIME

the roll number : 45 registration no is 78
enter the name of the person do you want
to delete
nt

record found

the records after deleting: name nt age: 45 roll
number: 45 registration number: 12

the records after deleting: name: ty age: 45
roll number: 45 registration number: 78
enter the number of records do you want to
enter for employee
3

enter the name and age of the person
name:

gh

age:

45

name:

fdg

age:

78

name:

ui

age:

23

enter the salary and profile of the employee
salary:

Experiment:

Date _____
Page No. 23

{

if (ne.equals(name[i]))

{

ai;

for (int j = a; j < n - 1; j++)

{

name[j] = name[j + 1];

age[j] = age[j + 1];

roll[j] = roll[j + 1];

reg[j] = reg[j + 1];

salary[j] = salary[j + 1];

prof[j] = prof[j + 1];

{ count++;

break;

}

if (count > 0)

{

System.out.println ("record found");

}

else

{

System.out.println ("record not found");

}

}

class student extends person

{

CLASSTIME

45

profile:

iu

salary:

53

profile:

jg

salary:

56

profile:

yr

the name is gh and age is 45

the name is fdg and age is 78

the name is ui and age is 23

the salary: 45 profile of the employee is: iu

the salary: 53 profile of the employee is: jg

the salary: 56 profile of the employee is: yr

enter the name of the person do you want

to delete

ui

record found

records after delete: name: gh age: 45

salary: 45 profile: iu

records after delete: name: fdg age: 78

salary: 53 profile: jg

BUILD SUCCESSFUL (total time: 1 minute
9 seconds)

Experiment:

Date _____
Page No. 24

void read1()

{

super.read();

System.out.println("enter the roll number and
registration number of the student");

for(int i=0; i<n; i++)

{

System.out.println("roll number");

roll[i]=sc.nextInt();

System.out.println("registration number");

reg[i]=sc.nextInt();

}

}

void show1()

{

for(int i=0; i<n; i++)

{

System.out.println("the roll number:" + roll[i] +

"registration no. is " + reg[i]);

}

void delete1()

{

super.delete();

for(int i=0; i<n; i++)

{

System.out.println("the records after deleting:
name: " + (name[i]) + "age: " + (age[i]) + "roll number: "

CLASSTIME

+ (roll[i]) + "registration number :" + (reg[i]));
 }
}

public class Java Application 27 {

 public static void main(String[] args) {

 System.out.println ("enter the number of
seconds do you want to enter for students");

 student st = new student();

 st.read1();

 st.show1();

 st.delete1();

 System.out.println ("enter the number of
records do you want to enter for employee");

 employee em = new employee();

 em.read2();

 em.show2();

 em.delete2();

 }

}

Output:

run:

enter the radius of a circle

5

the area of a circle is 78.53981633974483
and circumference 31.41592653589793

enter the side of the square

7

the area of a square is 49.0 and volume
is 343.0

enter the height and radius of the cylinder

4

6

the area of the cylinder is 376.99111843077515
and volume is 75.39822368615503

enter the radius of the sphere

5

the area of the sphere is 314.1592653589793
and volume is 392.6990816987241

enter the side of the cube

2

Experiment:

Practical - 7

Date _____

Page No. 1

Using the concept of multiple inheritance create classes : Shape, Circle, Square, Cube, Sphere, Cylinder. Your classes may only have the class variable specified in the table below and the methods Area and /or Volume to output their area and /or volume.

Sol.

```
package javaapplication21;  
import java.util.Scanner;  
class Shape
```

```
{
```

```
String name;
```

```
Shape()
```

```
{
```

```
}
```

```
}
```

```
class Circle extends Shape
```

```
{
```

```
double radius;
```

```
Circle (double r, String n)
```

```
{
```

```
radius = r;
```

```
name = n; }
```

```
double area()
```

```
{
```

CLASSTIME

the area of cube is 24.0 and volume is 8.0
BUILD SUCCESSFUL (total time 6 minutes.
20 seconds)

Experiment:

Date _____
Page No. 2

```
return (Math.PI * radius * radius);  
}  
double circumference()  
{  
    return (2 * Math.PI * radius);  
}  
  
class Square extends Shape  
{  
    double side;  
    Square (double s, string n)  
    {  
        side = s;  
        name = n;  
    }  
    double area()  
    {  
        return (side * side);  
    }  
    double volume()  
    {  
        return (Math.pow(side, 3));  
    }  
  
class Cylinder extends Circle  
{  
    double height;
```

CLASSTIME

Experiment :

Date _____
Page No. 3

Cylinder (double r, double h, String n) {

super (r, n);

radius = r;

name = n;

height = h;

}

double area ()

{

return (2 * Math.PI * radius * height) + (2 * Math.PI

* Math.pow (radius, 2));

}

double volume ()

{

return (Math.PI * radius * height);

}

}

class Sphere extends Circle

{

Sphere (double r, String n) {

super (r, n);

radius = r;

name = n;

}

double area ()

{

return (4 * Math.PI * Math.pow (radius, 2));

}

Experiment :

Date _____
Page No. 4

```
double volume ()  
{
```

```
    return ((4/3) * Math.PI * Math.pow (radius, 3));  
}
```

```
}
```

```
class Cube extends Square
```

```
{
```

```
Cube (double s, String n) {
```

```
    super (s, n);
```

```
    side = s;
```

```
    name = n;
```

```
}
```

```
double area ()
```

```
{
```

```
    return (6 * Math.pow (side, 2));
```

```
}
```

```
double volume ()
```

```
{
```

```
    return (Math.pow (side, 3));
```

```
}
```

```
}
```

```
public class Java Application 21 {
```

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

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

```
    System.out.println ("enter the radius of a  
circle");
```

```
    int a = sc.nextInt();
```

Experiment:

Date _____
Page No. 5

Circle c = new Circle (a, "circle");

System.out.println ("the area of a square
is " + sq.area () + " and volume is " + c.
circumference ());

System.out.println ("enter the side of a
square");

int b = sc.nextInt();

Square sq = new Square (b, "square");

System.out.println ("the area of a square
is " + sq.area () + " and volume is " +
sq.volume());

System.out.println ("enter the height and
radius of the cylinder");

int e = sc.nextInt();

int f = sc.nextInt();

Cylinder cy = new Cylinder (e, f,
"cylinder");

System.out.println ("the area of the
cylinder is " + cy.area () + " and volume is "
+ cy.volume());

System.out.println ("enter the radius of
the sphere");

int g = sc.nextInt();

Sphere sp = new Sphere (g, "sphere");

System.out.println ("the area of the sphere is ",
+ sp.area () + " and volume is " + sp.volume());

System.out.println ("enter the side of the cube");

Experiment :

Date _____
Page No. 6

int h = sc.nextInt();

Cube cu = new Cube(h, "cube");

System.out.println("the area of cube is " +
cu.area() + " and volume is " + cu.volume());

}

}

Output:

run:

name is : john date of birth : 17/1/20 major: hum
student id: 141

name is: admin date of birth : 20/8/10 salary:
20000 subject: maths

Experiment:

Practical - 8

Date _____
Page No. 7

Write a program to create class Person.

i) Make two classes , Student and Instructor , inherit from person. A person has a name and year of birth.

ii) A student has a major , student Id .

iii) An instructor has salary , subject .

Write the class definitions , the constructors, set methods , get methods and for all classes .

Sol.

```
package javaapplication28;  
import java.util.Scanner;  
class person  
{  
    private String name;  
    private String dob;
```

```
    public String getName() {  
        return name;  
    }
```

```
    public String getDob() {  
        return dob;  
    }
```

```
    public void setName (String name) {  
        this.name = name;
```

CLASSTIME

Experiment :

Date _____

Page No. 8

}

```
public void setDob (String dob) {  
    this.dob = dob;  
}
```

}

```
class student extends person  
{
```

```
private int sid;
```

```
private String major;
```

```
public int get Sid() {  
    return sid;
```

}

```
public void set Sid (int sid) {  
    this.sid = sid;
```

}

```
public String get Major () {  
    return major;
```

}

```
public void set Major (String Major) {  
    this.major = major;
```

}

}

```
class instructor extends person
```

{

```
private int Salary;
```

```
private String subj;
```

Experiment:

Date _____
Page No. 9

```
public int getSalary() {  
    return salary;  
}  
  
public void setSalary (int salary) {  
    this.salary = salary;  
}  
  
public String getSubj() {  
    return subj;  
}  
  
public void setSubj (String subj) {  
    this.subj = subj;  
}  
  
}  
  
public class JavaApplication28 {  
    public static void main (String [] args) {  
        Student st = new Student ();  
        st.setName ("John");  
        st.setDob ("17/1/20");  
        st.setMajor ("hum");  
        st.setSid (141);  
  
        System.out.println ("name is :" +  
            st.getName () + " date of birth :" + st.getDob () +  
            " major :" + st.getMajor () + " student id :" +  
            st.getSid ());  
  
        Instructor it = new Instructor ();  
        it.setName ("admin");  
        it.setDob ("20/8/10");
```

Experiment :

Date _____
Page No. 10

```
it.setSalary(20000);
it.setSubj("maths");
System.out.println("name is :" +
it.getName() + " date of birth :" +
it.getDob() + " salary :" + it.getSalary() +
"subject :" + it.getSubj());
```

3

Output :

```
Animal name 1 : Cow
Animal sound 2 : chick
Animal sound : cluck
```

Old MacDonald had a farm and several types of animals. Every animal shared certain characteristics: they had a type (such as cow, chick or pig) and each made a sound (moo, chick and oink). An Interface defines those things required to be an animal on the farm. Define new classes for the Old MacDonald that implement the animal and farm class. Create Array of object of animal to define the different types of animal in the farm. Also create appropriate methods to get and set the properties.

Sol.

```
package first;
interface animal
{
```

```
    public String gettype();
    public void settype (String x);
    public String getsound ();
    public void setsound (String x);
}
```

```
Class cow implements Animal
{
```

```
    String type , sound ;
    Public void settype (String x) .
```

Experiment :

Date _____
Page No. 12

 type = x ;
 }

 public String gettype ()

 return type ;
 }

 public void setsound (String x)

 sound = x ;
 }

 Public String getSound ()

 {

 return sound ;

 }

Class chick implements Animal

{

 String type , sound ;

 public void settype (String x)

{

 type = x ;
 }

 public String gettype ()

{

 sound = x ;
 }

 Public String getsound ()

{

Return sound;

}

?

Public class macdonald

{

Public static void main (String [] args)

{

// TODO Auto-generated method stub

Animal c [] = new Animal [2];

c [0] = new cow();

c [0].settype ("cow");

c [0].setsound ("moo");

c [1] = new chick();

c [1].settype ("chick");

c [1].setsound ("chuck");

for (int i=0; i<2; i++)

{

System.out.println ("Animal name " + (i+1) +
"; " + c [i].gettype());System.out.println ("Animal sound :" + c [i].
getsound());

}

?

Output:

run:

enter the number of records do you want to enter
for the engineering students.
3

enter the name and student id of the
engineering students
name:

wrg
student id:

757

name:

yr

student id:

56

name:

tr

student id:

75

the records you entered for the engineering
students are

name : [wrg, yr, tr] student id : [757, 56, 75]

Experiment :

Practical - 10

Date _____

Page No. 14

Write a program with Student as abstract
class and create derive classes Engineering,
Medicine and Science from base class
Student. Create the objects of the
derived classes and process them and
access them using array of pointer of
type base class Student.

Sol.

```
package javaapplication29;  
import java.util.Arrays;  
import java.util.Scanner;
```

```
abstract class Student
```

```
{
```

```
Scanner sc = new Scanner (System.in);  
int n = sc.nextInt();
```

```
String name [] = new String [n];  
int sid [] = new int [n];
```

```
abstract void read();
```

```
abstract void show();
```

```
}
```

```
class engineering extends Student
```

```
{
```

② Override

CLASSTIME

enter the number of records do you want to
enter for the medical students
3

enter the name and student id of the medical
students
name:
aty
student id:
785

name:
ytnt
student id:
87

name:
tyt
student id:
75

the records you entered for the medical
students are

name: [aty, ytnt, tyt] sid: [786, 87, 76]

enter the number of records do you want
to enter for the science students
3

enter the name and student id of the
science students
name:
rey

Experiment:

Date _____
Page No. 15

```
void read() {  
    System.out.println("enter the name and  
student id of the engineering student");  
    for (int i = 0; i < n; i++) {  
        System.out.println("name:");  
        name[i] = sc.nextLine();  
        System.out.println("student id:");  
        sid[i] = sc.nextInt();  
    }  
  
    void show() {  
        System.out.println("the records you  
entered for the engineering students are");  
        System.out.println("name:" + Arrays.  
        toString(name) + " student id:" + Arrays.  
        toString(sid));  
    }  
  
    class Medical extends student {  
        @Override  
        void read() {  
            System.out.println("enter the name  
and student id of the medical students");  
            for (int i = 0; i < n; i++) {  
                CLASSTIME
```

```
student id:  
78  
name:  
yuy  
student id:  
75  
name:  
tuyu  
student id:  
765  
the records you entered for the science  
students  
name: [ray, yuy, tuyu] student id: [78, 75, 765]  
BUILD SUCCESSFUL (total item: 27 seconds)
```

Experiment:

Date _____
Page No. 16

```
System.out.println("name");  
name[i] = sc.nextInt();  
System.out.println("student id");  
sid[i] = sc.nextInt();  
}
```

}

② Override

```
void show() {
```

```
System.out.println("the records you  
entered for the medical students are ");
```

```
System.out.println("name:" + Arrays.  
toString(name) + "sid:" + Arrays.toString(sid));  
}
```

}

```
class science extends student
```

{

③ Override

```
void read() {
```

```
System.out.println("enter the name and  
student id of the science students ");
```

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

{

```
System.out.println("name");  
name[i] = sc.nextInt();
```

```
System.out.println("student id");  
sid[i] = sc.nextInt();  
}
```

CLASSTIME

3

@ Override

```
void show() {
```

```
    System.out.println("the records you  
entered for the science students");
```

```
    System.out.println("name:" + Arrays.  
toString(name) + "student id:" + Arrays.  
toString(sid));
```

3

3

```
public class JavaApplication29 {
```

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

```
        System.out.println("enter the number  
of records do you want to enter for the  
engineering students");
```

```
        engineering en = new engineering();  
        en.read();  
        en.show();
```

```
        System.out.println("enter the number  
of records do you want to enter for the  
medical students");
```

```
        medical md = new medical();
```

```
        md.read();
```

```
        md.show();
```

```
        System.out.println("enter the number  
of records do you want to enter for the  
science students");
```

Experiment :

Date _____

Page No. 18

science sc = new science();

sc.read();

sc.show();

}

}