1. With Class diagram Create a class called Birthday with attributes Date, month ,year and create another class called person with its attributes name of the person ..Create Has-A Relation ship between the class using Aggregation in order to print the details by ...(i) Using Setter ,Getter ,toString Method

(ii) Without using Getter ,toString Method

(iii)using constructor , Getter ,toString method

//Birthday

**package** Niranjan;

**public** **class** Birthday {

**private** **int** month;

**private** **int** day;

**private** **int** year;

Birthday(**int** month,**int** day,**int** year)

{

**this**.month=month;

**this**.day=day;

**this**.year=year;

}

**public** **int** getM()

{

**return** month;

}

**public** **int** getD()

{

**return** day;

}

**public** **int** getY()

{

**return** year;

}

**public** String toString()

{

String s=String.*format*("Month=%d%n Day=%d%s year=%d%s",getM(),getD(),getY());

**return** s;

}

}

//Person

**package** Niranjan;

**import** java.util.Scanner;

**public** **class** Person {

String name;

Birthday per;

//Has-A relationship

Person(String name,Birthday per)

{

**this**.name=name;

**this**.per=per;

}

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

Birthday p=**new** Birthday(7,11,2003);

Person n=**new** Person("Niranjan",p);

System.***out***.println(n.name);

System.***out***.println(n.per.getD());

System.***out***.println(n.per.getM());

System.***out***.println(n.per.getY());

}

}

2.) With Class diagram Create a class called Author with attributes author name and author age and call the corresponding address constructor by passing the arguments and create another class called Book with its attributes Book name and book id ..Create Has-A Relation ship between the class using Aggregation in order to print the details of author and books.....(i) Using Getter ,toString Method (ii) Without using Getter ,toString Method

**//Author Class**

**package** Sah;

**public** **class** Author {

**private** String Aname;

**private** **int** Aage;

Author(String Aname,**int** Aage)

{

**this**.Aname=Aname;

**this**.Aage=Aage;

}

**public** String getN()

{

**return** Aname;

}

**public** **int** getA()

{

**return** Aage;

}

**public** String toString()

{

String s=String.*format*("Author name=%s%n Author age=%d",getN(),getA());

**return** s;

}

}

**//Book class**

**package** Sah;

**public** **class** Book {

String Bname;

String Bid;

Author A;

//Has-A relationship

Book(String Bname,String Bid,Author A)

{

**this**.Bname=Bname;

**this**.Bid=Bid;

**this**.A=A;

}

**public** String getN()

{

**return** Bname;

}

**public** String getid()

{

**return** Bid;

}

**public** String toString()

{

**return** String.*format*("Bname=%s%n Bid=%s%n",getN(),getid());

}

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

Author au=**new** Author("Aashutosh",22);

Book b=**new** Book(" Java Programming","21SC1101",au);

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

System.***out***.println(b.A.getN());

System.***out***.println(b.A.getA());

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

System.***out***.println(b.Bname);

System.***out***.println(b.Bid);

}

}

1. With Class diagram Create a Cuboid class with 3 public instance variables length, breadth and height of type double,and a method volume (). Create 2 objects with different values obtained by command line arguments and print the volume of each. (The program must take 6 values as input)

//Cuboid

**package** Sc;

**public** **class** Cuboid {

**static** **double** *length*;

**static** **double** *breadth*;

**static** **double** *height*;

**static** **double** *l*;

**static** **double** *b*;

**static** **double** *h*;

**public** **double** Volume()

{

**return** *length*\**breadth*\**height*;

}

**public** String toString()

{

String s=String.*format*("Volume =%s%n ",Volume());

**return** s;

}

**public** **static** **void** main(String[] args)

{

Cuboid c=**new** Cuboid();

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

*length*=Double.*parseDouble*(args[0]);

*breadth*=Double.*parseDouble*(args[1]);

*height*=Double.*parseDouble*(args[2]);

*l*=Double.*parseDouble*(args[3]);

*b*=Double.*parseDouble*(args[4]);

*h*=Double.*parseDouble*(args[5]);

**double** volume= *l*\**b*\**h*;

System.***out***.println("Volume Of Cuboid is:" +volume);

System.***out***.println(c);

}

}

4.) With Class diagram Write a Java Program to create Student class with ID, name, gender, and branch. Use getter and setters. The ID must be 9-digit number, name must not have special characters and digits, gender must be either M/F and branch must be either ECE/CSE/ME/ECSE/CE/BT/EEE. Use toString () to format the details of Student. Read data from console and create 10 student objects and print the data of each student.

//Student class

**package** St;

**import** java.util.Scanner;

**public** **class** Student {

**private** String ID;

**private** String Name;

**private** String Gender;

**private** String Branch;

**public** **void** setID(String ID)

{

**this**.ID=ID;

}

**public** **void** setName(String Name)

{

**this**.Name=Name;

}

**public** **boolean** setGender(String Gender)

{

**if**(Gender.matches("[Male]+") || Gender.matches("[Female]+"))

{

**this**.Gender=Gender;

**return** **true**;

}

**return** **false**;

}

**public** **boolean** setBranch(String Branch)

{

**if**(Gender.equals("[ECE]+") || Gender.equals("[CSE]+")|| Gender.equals("[ME]+")|| Gender.equals("[CS&IT]+")|| Gender.equals("[EEE]+")|| Gender.equals("[CE]+")|| Gender.equals("[BT]+")|| Gender.equals("[IOT]+"))

{

**this**.Branch=Branch;

**return** **true**;

}

**return** **false**;

}

**public** String getID()

{

**return** ID;

}

**public** String getName()

{

**return** Name;

}

**public** String getGENDER()

{

**return** Gender;

}

**public** String getBranch()

{

**return** Branch;

}

**public** String toString()

{

String s=String.*format*("Student ID=%s%n Student name=%s%n Student Gender=%s%n Student Branch=%s%n",getID(),getName(),getGENDER(),getBranch());

**return** s;

}

**public** **static** **void** main(String[] args)

{

Student s=**new** Student();

Scanner ss=**new** Scanner(System.***in***);

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

{

System.***out***.println(">>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>");

System.***out***.println("Enter the student ID");

s.setID(ss.nextLine());

System.***out***.println("Enter the student Name");

s.setName(ss.nextLine());

System.***out***.println("Enter the student Gender(Male/Female)");

s.setGender(ss.next());

System.***out***.println("Enter the student Branch");

s.setBranch(ss.next());

}

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

{

System.***out***.println(">>>>>>>>>>>>>>>>>>>>>>>>>>>>>All Students Details>>>>>>>>>>>>>>>>>>>>>>>>>>>>>");

System.***out***.println(s);

}

}

}

5.) With Class diagram Create a class Address with instance variables city, state, and country of String type. create a parameterized constructor and a toString (). Create a class Employee with instance variable id of type int, name type String, and address of type Address. Create a parameterized constructor and a toString (). And create 2 objects for the class Employee and print the details of each object in main () of different class.

**package** Stu;

**public** **class** Address {

**private** String City;

**private** String State;

**private** String Country;

Address(String City,String State,String Country)

{

**this**.City=City;

**this**.State=State;

**this**.Country=Country;

}

**public** String getCity()

{

**return** City;

}

**public** String getState()

{

**return** State;

}

**public** String getCountry()

{

**return** Country;

}

**public** String toString()

{

String s=String.*format*("City=%s%n State=%s%n Country=%s%n", getCity(),getState(),getCountry());

**return** s;

}

}

//employee

**package** Stu;

**import** java.util.Scanner;

**public** **class** Employee {

**int** ID;

String Name;

**static** Address *Add*;

Employee(**int** ID,String Name,Address Add)

{

**this**.ID=ID;

**this**.Name=Name;

Employee.*Add*=Add;

}

**public** **int** getID()

{

**return** ID;

}

**public** String getName()

{

**return** Name;

}

**public** String toString()

{

**return** String.*format*("ID=%d%n Name=%s%n",getID(),getName());

}

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

Scanner s=**new** Scanner(System.***in***);

System.***out***.println("Enter the City,State,Country");

Address a=**new** Address(s.nextLine(),s.nextLine(),s.nextLine());

System.***out***.println("Enter ID and name");

Employee e=**new** Employee(s.nextInt(),s.nextLine(),*Add*);

System.***out***.println(">>>>>>>>Personal Detail>>>>>>>>");

System.***out***.println(a);

System.***out***.println(">>>>>>>>Address Detail>>>>>>>>");

System.***out***.println(e);

}

}

6.) With Class diagram Overload the Cuboid class with all 3 types of constructors and create 3 objects each invoking a different type of constructor from the main method of Demo class. Enhance the above code by chaining the constructors

**package** p1;

**public** **class** Cuboid {

**private** **int** a;

**private** **int** b;

**private** **int** c;

//Default Constructor

Cuboid()

{

System.***out***.println("Default Constructor");

a=5;

b=6;

c=7;

}

//Parameterized Constructor

Cuboid(**int** x,**int** y,**int** z)

{

System.***out***.println("Parameterized Constructor");

a=x;

b=y;

c=z;

}

//copy constructor

Cuboid(Cuboid obj)

{

System.***out***.println("Copy Constructor");

a=obj.a;

b=obj.b;

c=obj.c;

}

**public** String toString1()

{

String st = String.*format*("A=%d%n B=%d%n",a,b);

**return** st;

}

**public** String toString()

{

String str = String.*format*("A=%d%nB=%d%n",a,b);

**return** str;

}

**public** **static** **void** main(String[] args)

{

Cuboid cu=**new** Cuboid();

System.***out***.println(cu.a+" "+cu.b+" "+cu.c+" ");

Cuboid c=**new** Cuboid(9,10,11);

System.***out***.println(c);

Cuboid obj=**new** Cuboid(9,10,11);

System.***out***.println(obj);

}

}