1.PERSON DETAILS PROGRAM:

class Person {

private String name;

private int age;

public void setName(String name) {

this.name=name;

}

public String getName() {

return name;

}

public void setAge(int age) {

this.age=age;

}

public int getAge() {

return age;

}

}

public class Main

{

public static void main(String[] args) {

Person person=new Person();

person.setName("Alice");

person.setAge(30);

System.out.println(person.getName()+ " is "+ person.getAge()+" years old");

}

}

// 2.area of circle program

import java.util.Scanner;

public class CircleArea {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

// Input radius

System.out.print("Enter the radius of the circle: ");

double radius = scanner.nextDouble();

// Calculate area

double area = Math.PI \* radius \* radius;

// Output result

System.out.println("The area of the circle is: " + area);

scanner.close();

}

}

// 3.STUDENT DETAILS PROGRAM

Class A{

String name=”Akshaya”;

int rollno=15;

String dept=”AI &DS”;

Char sec=’A’;

Void Student\_details()

{

String C=”Name: “ +name + “ “ +”rollno” + “Dept” + “ “ +dept + “Section: “+ “ “+ sec;

System.out.println(“The student details :”);

public class Student{

public statis void main(Strings[] args){

A obj= new A();

Obj.Student\_detail():

}

//4.EXAMPLE PROGRAM FOR METHOD OVERLOADING:

**package** akshaya;

**public** **class** aksh

{

**int** a;

String name ,S;

**void** set(**int** a)

{

**this**.a=a;

System.***out***.println("The value of a is " +a);

}

**void** set(**int** a, String name)

{

**this**.a=a;

**this**.name=name;

System.***out***.println("The value of a is " + a + "name is " + name);

}

**void** set(String name,String S)

{

**this**.name=name;

**this**.S=S;

System.***out***.println("The value of name "+ name + S);

}

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

{

aksh obj=**new** aksh();

obj.set("Same","Vel");

obj.set(9,"Vel");

obj.set(8);

}

}

//OUTPUT:

The value of name SameVel

The value of a is 9name is Vel

The value of a is 8

5.RUN TIME POLYMORPHISM:

class B {

public void get() {

System.out.println("This is super class");

}

}

class C extends B {

@Override

public void get() {

System.out.println("This is subclass");

}

}

public class Main {

public static void main(String[] args) {

C obj = new C();

obj.get();

}

}

//OUTPUT:

This is subclass

6.

**class B {**

**public void get() {**

**System.out.println("This is super class");**

**}**

**}**

**class C extends B {**

**@Override**

**public void get() {**

**System.out.println("This is subclass");**

**}**

**}**

**public class Main {**

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

**B obj = new C();**

**obj.get();**

**}**

**}**

**//OUTPUT:**

**This is subclass**

**ASSIGNMENT QUESTION**

1.Define classes for books authors and members .implement basic opertations using OOP concepts

class Author {

private String name;

private String bio;

public Author(String name, String bio) {

this.name = name;

this.bio = bio;

}

public String getName() {

return name;

}

public String getBio() {

return bio;

}

public void displayAuthor() {

System.out.println("Author: " + name + " | Bio: " + bio);

}

}

class Book {

private String title;

private Author author;

private boolean isAvailable;

public Book(String title, Author author) {

this.title = title;

this.author = author;

this.isAvailable = true; // by default, book is available

}

public String getTitle() {

return title;

}

public Author getAuthor() {

return author;

}

public boolean isAvailable() {

return isAvailable;

}

public void borrowBook() {

if (isAvailable) {

isAvailable = false;

System.out.println("You borrowed: " + title);

} else {

System.out.println(title + " is currently not available.");

}

}

public void returnBook() {

isAvailable = true;

System.out.println("You returned: " + title);

}

public void displayBook() {

System.out.println("Book: " + title + " | Author: " + author.getName() + " | Available: " + isAvailable);

}

}

class Member {

private String name;

private int memberId;

private Book borrowedBook;

public Member(String name, int memberId) {

this.name = name;

this.memberId = memberId;

this.borrowedBook = null;

}

public void borrow(Book book) {

if (borrowedBook == null && book.isAvailable()) {

borrowedBook = book;

book.borrowBook();

} else {

System.out.println(name + " already borrowed a book or the book is not available.");

}

}

public void returnBook() {

if (borrowedBook != null) {

borrowedBook.returnBook();

borrowedBook = null;

} else {

System.out.println("No book to return.");

}

}

public void displayMember() {

System.out.println("Member: " + name + " | ID: " + memberId);

if (borrowedBook != null) {

System.out.println("Borrowed: " + borrowedBook.getTitle());

} else {

System.out.println("No book borrowed.");

}

}

}

public class LibrarySystem {

public static void main(String[] args) {

Author a1 = new Author("J.K. Rowling", "British author, famous for Harry Potter");

Book b1 = new Book("Harry Potter and the Sorcerer's Stone", a1);

Member m1 = new Member("Alice", 101);

b1.displayBook();

a1.displayAuthor();

m1.displayMember();

m1.borrow(b1);

m1.displayMember();

m1.returnBook();

m1.displayMember();

}

}

//CONSTRUCTOR

Class Box;

{  
double w,h,d;

Box(int a, int b, int c);

{  
System.out.println(“this is a constructor”);

W=a;

H=b;

D=c;

System.out.println(“The values of w, h,d are “+w+h+d);

}

}

Public static void main(String[] args)

{  
Box My box=new Box(5,10,15);

}

}

//CONSTRUCTOR OVERLOADING

class Box {

double width, height, depth;

// Default constructor

Box() {

width = 1;

height = 1;

depth = 1;

System.out.println("Default constructor called");

}

// Parameterized constructor with 3 values

Box(double w, double h, double d) {

width = w;

height = h;

depth = d;

System.out.println("Parameterized constructor called");

}

// Constructor with one value (cube)

Box(double side) {

width = height = depth = side;

System.out.println("Cube constructor called");

}

void displayVolume() {

double volume = width \* height \* depth;

System.out.println("Volume: " + volume);

}

}

public class Main {

public static void main(String[] args) {

Box box1 = new Box();

box1.displayVolume();

Box box2 = new Box(10, 5, 2);

box2.displayVolume();

Box box3 = new Box(4);

box3.displayVolume();

}

}