## 

## 

## 

## 

## 

## LAB 05

## INHERITANCE

EXERCISE 1

create a class called College with attribute String name, constructor to initialize the name attribute , a method called Admitted(). Create a subclass called CSE that extends Student class, with department attribute , Course() method to sub class. Print the details of the Student.

College:

String collegeName;

public College() { }

public admitted() { }

Student:

String studentName;

String department;

public Student(String collegeName, String studentName,String depart) { }

public toString()

Expected Output:

A student admitted in REC  
CollegeName : REC  
StudentName : Venkatesh  
Department : CSE

**For example:**

| **Result** |
| --- |
| A student admitted in REC  CollegeName : REC  StudentName : Venkatesh  Department : CSE |

**Answer:**

class College

{

protected String collegeName;

public College(String collegeName) {

this.collegeName = collegeName;

}

public void admitted() {

System.out.println("A student admitted in "+collegeName);

}

}

class Student extends College{

String studentName;

String department;

public Student(String collegeName, String studentName,String depart) {

super(collegeName);

this.studentName = studentName;

this.department = depart;

}

public String toString(){

return collegeName;

}

public String toString1(){

return studentName;

}

public String toString2(){

return department;

}

}

public class Main {

public static void main (String[] args) {

Student s1 = new Student("REC","Venkatesh","CSE");

s1.admitted();

System.out.println("CollegeName : " +s1.toString());

System.out.println("StudentName : "+s1.toString1());

System.out.println("Department : "+s1.toString2());

}

}

|  | **Expected** | **Got** |  |
| --- | --- | --- | --- |
|  | A student admitted in REC  CollegeName : REC  StudentName : Venkatesh  Department : CSE | A student admitted in REC  CollegeName : REC  StudentName : Venkatesh  Department : CSE |  |

EXERCISE 2

Create a class known as "BankAccount" with methods called deposit() and withdraw().

Create a subclass called SavingsAccount that overrides the withdraw() method to prevent withdrawals if the account balance falls below one hundred.

**For example:**

| **Result** |
| --- |
| Create a Bank Account object (A/c No. BA1234) with initial balance of $500:  Deposit $1000 into account BA1234:  New balance after depositing $1000: $1500.0  Withdraw $600 from account BA1234:  New balance after withdrawing $600: $900.0  Create a SavingsAccount object (A/c No. SA1000) with initial balance of $300:  Try to withdraw $250 from SA1000!  Minimum balance of $100 required!  Balance after trying to withdraw $250: $300.0 |

class BankAccount {

// Private field to store the account number

private String accountNumber;

// Private field to store the balance

private double balance;

// Constructor to initialize account number and balance

BankAccount(String accountNumber , double balance){

this.accountNumber = accountNumber;

this.balance = balance;

}

// Method to deposit an amount into the account

public void deposit(double amount) {

// Increase the balance by the deposit amount

balance+=amount;

System.out.println("New balance after depositing $1000: $"+balance);

}

// Method to withdraw an amount from the account

public void withdraw(double amount) {

// Check if the balance is sufficient for the withdrawal

if (balance >= amount) {

// Decrease the balance by the withdrawal amount

balance -= amount;

} else {

// Print a message if the balance is insufficient

System.out.println("Insufficient balance");

}

}

// Method to get the current balance

public double getBalance() {

// Return the current balance

return balance;

}

}

class SavingsAccount extends BankAccount {

// Constructor to initialize account number and balance

public SavingsAccount(String accountNumber, double balance) {

// Call the parent class constructor

super(accountNumber, balance);

}

// Override the withdraw method from the parent class

@Override

|  | **Expected** | **Got** |  |
| --- | --- | --- | --- |
|  | Create a Bank Account object (A/c No. BA1234) with initial balance of $500:  Deposit $1000 into account BA1234:  New balance after depositing $1000: $1500.0  Withdraw $600 from account BA1234:  New balance after withdrawing $600: $900.0  Create a SavingsAccount object (A/c No. SA1000) with initial balance of $300:  Try to withdraw $250 from SA1000!  Minimum balance of $100 required!  Balance after trying to withdraw $250: $300.0 | Create a Bank Account object (A/c No. BA1234) with initial balance of $500:  Deposit $1000 into account BA1234:  New balance after depositing $1000: $1500.0  Withdraw $600 from account BA1234:  New balance after withdrawing $600: $900.0  Create a SavingsAccount object (A/c No. SA1000) with initial balance of $300:  Try to withdraw $250 from SA1000!  Minimum balance of $100 required!  Balance after trying to withdraw $250: $300.0 |  |

EXERCISE 3

Create a class Mobile with constructor and a method basicMobile().

Create a subclass CameraMobile which extends Mobile class , with constructor and a method newFeature().

Create a subclass AndroidMobile which extends CameraMobile, with constructor and a method androidMobile().

display the details of the Android Mobile class by creating the instance. .

class Mobile{  
  
}  
class CameraMobile extends Mobile {

}

class AndroidMobile extends CameraMobile {

}

expected output:

Basic Mobile is Manufactured  
Camera Mobile is Manufactured  
Android Mobile is Manufactured  
Camera Mobile with 5MG px  
Touch Screen Mobile is Manufactured

**For example:**

| **Result** |
| --- |
| Basic Mobile is Manufactured  Camera Mobile is Manufactured  Android Mobile is Manufactured  Camera Mobile with 5MG px  Touch Screen Mobile is Manufactured |

**Answer:**

class Mobile{

Mobile(){

System.out.println("Basic Mobile is Manufactured");

}

void basicMobile(){

System.out.println("Basic Mobile");

}

}

class CameraMobile extends Mobile{

CameraMobile(){

System.out.println("Camera Mobile is Manufactured");

}

void newfeatures(){

System.out.println("Camera Mobile with 5MG px");

}

}

class AndroidMobile extends CameraMobile{

AndroidMobile(){

System.out.println("Android Mobile is Manufactured");

}

void androidmobile(){

System.out.println("Touch Screen Mobile is Manufactured");

}

}

public class Main{

public static void main(String[] args){

AndroidMobile a = new AndroidMobile();

a.newfeatures();

a.androidmobile();

}

}

|  | **Expected** | **Got** |  |
| --- | --- | --- | --- |
|  | Basic Mobile is Manufactured  Camera Mobile is Manufactured  Android Mobile is Manufactured  Camera Mobile with 5MG px  Touch Screen Mobile is Manufactured | Basic Mobile is Manufactured  Camera Mobile is Manufactured  Android Mobile is Manufactured  Camera Mobile with 5MG px  Touch Screen Mobile is Manufactured |  |