**Assignment No- 5**

1. Create a base class BankAccount with methods like deposit() and withdraw(). Derive a class SavingsAccount that overrides the withdraw() method to impose a limit on the withdrawal amount. Write a program that demonstrates the use of overridden methods and proper access modifiers & return the details.

**package** org.example.bankaccount;

**import** java.util.Scanner;

**class** bankAccount{

**protected** String accountHolder;

**protected** **double** balance;

**public** bankAccount(String accountHolder, **double** balance) {

**this**.accountHolder = accountHolder;

**this**.balance = balance;

}

**public** **void** deposit(**double** amount) {

**if**(amount>=0) {

balance = balance + amount;

System.***out***.println("Deposited : "+amount);

}

**else** {

System.***out***.println("Invalid deposit amount : ");

}

}

**public** **void** withdraw(**double** amount) {

**if**(amount >0 && amount <= balance) {

balance = balance - amount;

System.***out***.println("Withdrwal amount : "+amount);

}

**else** {

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

}

}

**public** **double** getBalance() {

**return** balance;

}

**public** String toString() {

**return** "Account Holder :"+accountHolder+" Balance Amount :"+balance;

}

}

**class** savingAccount **extends** bankAccount{

**private** **double** withdrawLimit;

**public** savingAccount(String accountHolder, **double** balance, **double** withdrawLimit) {

**super**(accountHolder, balance);

// **TODO** Auto-generated constructor stub

**this**.withdrawLimit=withdrawLimit;

}

@Override

**public** **void** withdraw(**double** amount) {

**if**(amount>0 && amount<=balance && amount<=withdrawLimit) {

balance -= amount;

System.***out***.println("Withdraw : "+ amount);

}

**else** **if**(amount > withdrawLimit) {

System.***out***.println("Withdrawl Denied");

}

**else** {

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

}

}

**public** **double** getBalance(){

**return** balance;

}

@Override

**public** String toString() {

**return** **super**.toString() +" , Withdrawal Amount : " + withdrawLimit;

}

}

**public** **class** Program1 {

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

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

//Data from user input

System.***out***.println("Enter the account holder name : ");

String basicHolder = sc.nextLine();

System.***out***.println("Enter the initial balance");

**double** basicBalance = sc.nextDouble();

//object creation

bankAccount bank = **new** bankAccount(basicHolder,basicBalance);

System.***out***.println("Basic Account created :");

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

//deposit amount

System.***out***.println("Enter the amount to be deposited in basic amount :");

**double** depositAmount=sc.nextDouble();

bank.deposit(depositAmount);

//bank.withdraw(2000);

//withdraw amount

System.***out***.println("Enter the amount to withdraw from bank account :");

**double** withdrawAmount=sc.nextDouble();

bank.withdraw(withdrawAmount);

System.***out***.println("Updated Balance : "+bank.getBalance());

sc.nextLine();

System.***out***.println("Enter the account Holder name :");

String savingsHolder = sc.nextLine();

System.***out***.println("Enter initial balance amount for saving account :");

**double** savingsBalance = sc.nextDouble();

System.***out***.println("Enter the withdrawLimit :");

**double** withdrawLimit = sc.nextDouble();

savingAccount savingsAccount = **new** savingAccount(savingsHolder,savingsBalance,withdrawLimit);

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

System.***out***.println("Enter the amount to withdraw :");

**double** savewithdraw=sc.nextDouble();

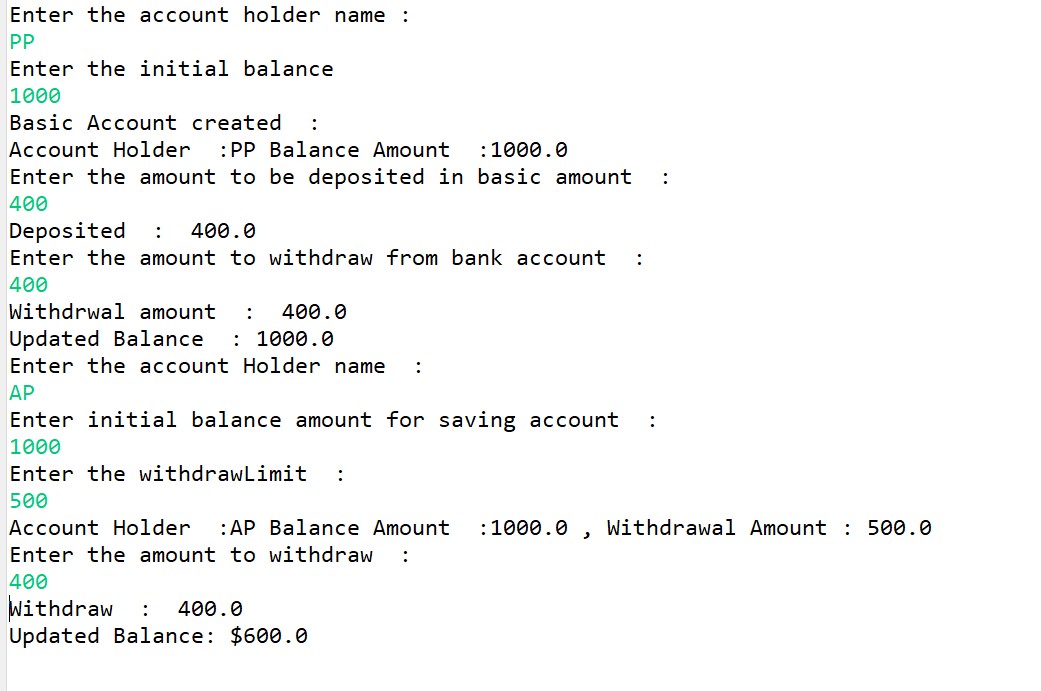
savingsAccount.withdraw(savewithdraw);

System.***out***.println("Updated Balance: $" + savingsAccount.getBalance());

sc.close();

}

}



2. Create a base class Vehicle with attributes like make and year. Provide a constructor in Vehicle to initialize these attributes. Derive a class Car that has an additional attribute model and write a constructor that initializes make, year, and model. Write a program to create a Car object and display its details.

**package** org.example.Vehicle;

**import** java.util.Scanner;

**class** Vehicle{

String make;

**int** year;

**public** Vehicle() {

}

**public** **void** acceptRecord() {

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

System.***out***.println("Enter the Company name : ");

**this**.make=sc.nextLine();

System.***out***.println("Enter the Year : ");

**this**.year=sc.nextInt();

sc.nextLine();

}

**public** **void** displayDetails() {

System.***out***.println("Make : "+ **this**.make);

System.***out***.println("Year : "+ **this**.year);

}

}

**class** Car **extends** Vehicle{

String model;

**public** Car() {

**super**();

}

**public** **void** acceptRecord() {

**super**.acceptRecord();

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

System.***out***.println("Enter Model : ");

**this**.model= sc.nextLine();

}

**public** **void** displayDetails() {

**super**.displayDetails();

System.***out***.println("Model : "+**this**.model);

}

}

**public** **class** Program1{

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

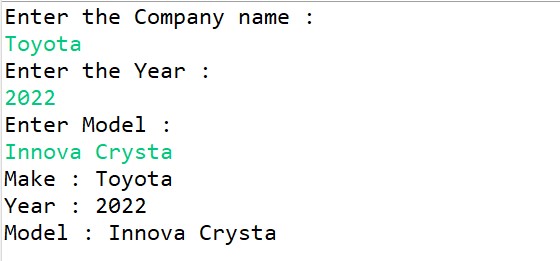
Car car = **new** Car();

car.acceptRecord();

car.displayDetails();

}

}



3.Create a base class Animal with attributes like name, and methods like eat() and sleep(). Create a subclass Dog that inherits from Animal and has an additional method bark(). Write a program to demonstrate the use of inheritance by creating objects of Animal and Dog and calling their methods.

**package** org.example.Animal;

**class** Animal{

String name;

**public** Animal(String name) {

**this**.name=name;

}

**public** **void** eat() {

System.***out***.println(name + "is eating.");

}

**public** **void** sleep() {

System.***out***.println(name + "is sleeping.");

}

}

**class** Dog **extends** Animal{

**public** Dog(String name) {

**super**(name);

}

**public** **void** bark() {

System.***out***.println(name + "is barking.");

}

}

**class** Dog **extends** Animal{

**public** Dog(String name) {

**super**(name);

}

**public** **void** bark() {

System.***out***.println(name + "is barking.");

}

}

**public** **class** Program1{

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

Animal animal = **new** Animal("Generic Animal");

animal.eat();

animal.sleep();

Dog dog = **new** Dog("Buddy");

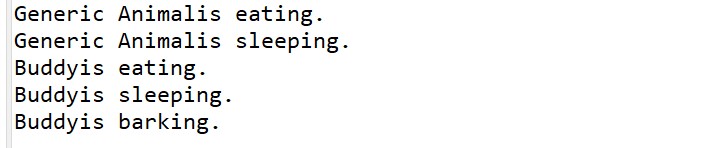
dog.eat();

dog.sleep();

dog.bark();

}

}



1. Build a class Student which contains details about the Student and compile and run its

instance.

**package** org.example.Student;

**import** java.util.Scanner;

**class** Student {

String name;

**int** stdid;

**double** grade;

**public** Student() {

}

**public** Student(String name, **int** stdid, **double** grade) {

**super**();

**this**.name = name;

**this**.stdid = stdid;

**this**.grade=grade;

}

**public** **void** acceptRecord() {

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

System.***out***.println("Enter Student Name : ");

**this**.name=sc.nextLine();

System.***out***.println("Enter StudentID : ");

**this**.stdid=sc.nextInt();

System.***out***.println("Enter Grade : ");

**this**.grade=sc.nextDouble();

}

**public** **void** printRecord() {

System.***out***.println("Name : "+ **this**.name);

System.***out***.println("Student ID : "+ **this**.stdid);

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

}

}

**public** **class** Program1{

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

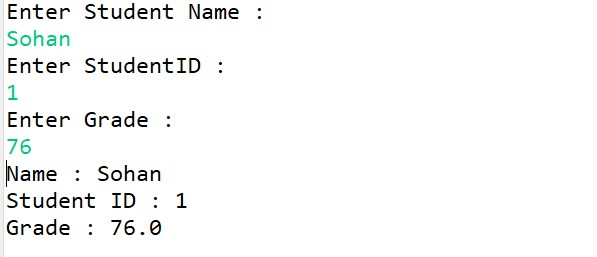
Student student = **new** Student();

student.acceptRecord();

student.printRecord();

}

}



5) Write a Java program to create a base class Vehicle with methods startEngine() and stopEngine(). Create two subclasses Car and Motorcycle. Override the startEngine() and stopEngine() methods in each subclass to start and stop the engines differently.

**package** org.example.CarsBikes;

**class** Vehicle{

**public** **void** startEngine() {

System.***out***.println("Starting the engine of the vehicle...");

}

**public** **void** stopEngine() {

System.***out***.println("Ending the engine of the vehicle...");

}

}

**class** Car **extends** Vehicle{

**public** **void** startEngine() {

System.***out***.println("Start the car engine with key ignition..");

}

**public** **void** stopEngine() {

System.***out***.println("Stopping the car engine with key ignition");

}

}

**class** Motorcycle **extends** Vehicle{

**public** **void** startEngine() {

System.***out***.println("Motorcycle starts with kickstart");

}

**public** **void** stopEngine() {

System.***out***.println("Motorcycle engine stops with key off");

}

}

**public** **class** Program1{

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

Vehicle car = **new** Car();

car.startEngine();

car.stopEngine();

Vehicle motorcycle = **new** Motorcycle();

motorcycle.startEngine();

motorcycle.stopEngine();

}

}

