1. Method Overloading: Write a class Calculator with overloaded methods add(). Implement add() methods that take:

- Two integers

- Two double values

- Three integers

- A variable number of integers

**CODE**

**package** code;

**public** **class** calculate {

**public** **int** add(**int** a, **int** b) {

**return** a + b;

}

**public** **double** add(**double** a, **double** b) {

**return** a + b;

}

**public** **int** add(**int** a, **int** b, **int** c) {

**return** a + b + c;

}

**public** **int** add(**int** numbers) {

**int** sum = 0;

**for** (**int** num : numbers) {

sum += num;

}

**return** sum;

}

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

calculate calculator = **new** calculate();

System.***out***.println("Sum of it: " + calculator.add(5, 3));

System.***out***.println("Sum of it: " + calculator.add(5.5, 3.2));

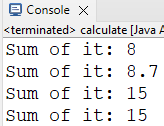
System.***out***.println("Sum of it: " + calculator.add(5, 3, 7));

System.***out***.println("Sum of it: " + calculator.add(1, 2, 3, 4, 5));

}

}

**OUTPUT**



2. Super Keyword: Create a class Person with a constructor that accepts and sets name and age. - Create a subclass Student that adds a grade property and initializes name and age using the super keyword in its constructor. - Demonstrate the creation of Student objects and the usage of super to call the parent class constructor.

**CODE**

**package** code;

**class** Person{

String name;

**int** age;

**public** Person(String name,**int** age) {

**this**.name = name;

**this**.age=age;

}

}

**class** student **extends** Person{

String grade;

**public** student(String name,**int** age,String grade) {

**super**(name,age);

**this**.grade=grade;

}

**public** **void** display() {

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

System.***out***.println("age"+age);

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

}

}

**public** **class** stud {

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

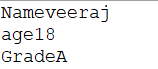
student stu1= **new** student("veeraj",18,"A");

stu1.display();

}

}

**OUTPUT**

****

3. Create a base class Shape with a method draw() that prints "Drawing Shape".

- Create a subclass Circle that overrides draw() to print "Drawing Circle".

- Inside the draw() method of Circle, call the draw() method of the Shape class using super.draw().

- Write a main method to demonstrate calling draw() on a Circle object.

**CODE**

**package** box;

**class** Shape {

**public** **void** draw() {

System.***out***.println("Drawing Shape");

}

}

**class** circle **extends** Shape{

**public** **void** draw() {

**super**.draw();

System.***out***.println("Drawing Circle");

}

}

**public** **class** rec {

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

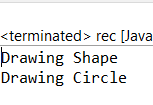
circle cir1=**new** circle();

cir1.draw();

}

}

**OUTPUT**



4. Create a base class BankAccount with a method deposit(amount) and a constructor that sets the initial balance.

- Create a subclass SavingsAccount that overrides deposit(amount) to add interest before depositing. Use the super keyword to call the deposit method of the base class.

- Write a main method to demonstrate creating a SavingsAccount and depositing an amount to see the effect of interest

**CODE**

**package** add;

**class** BankAccount {

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

**public** BankAccount(**double** initialBalance) {

**this**.balance = initialBalance;

}

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

balance += amount;

System.***out***.println("Deposited: " + amount + ", New Balance: " + balance);

}

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

**return** balance;

}

}

**class** SavingsAccount **extends** BankAccount {

**private** **double** interestRate;

**public** SavingsAccount(**double** initialBalance, **double** interestRate) {

**super**(initialBalance);

**this**.interestRate = interestRate;

}

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

**double** interest = amount \* interestRate / 100;

**double** amountWithInterest = amount + interest;

**super**.deposit(amountWithInterest);

}

}

**public** **class** bank {

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

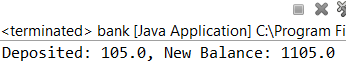
SavingsAccount savingsAccount = **new** SavingsAccount(1000, 5);

savingsAccount.deposit(100);

}

}

**OUTPUT**



5. Define a class Employee with properties name and salary and a method displayDetails().

- Create a subclass Manager that adds a property department and overrides displayDetails() to include department details. Use the super keyword to call the displayDetails() method of Employee within Manager.

- In the main method, create objects of Employee and Manager and call displayDetails() to show the details.

**CODE**

**package** code;

**class** employee{

**private** String name;

**private** **double** salary;

**public** employee(String name, **double** salary) {

**this**.name=name;

**this**.salary=salary;

}

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

System.***out***.println("Name: " + name + ", Salary: " + salary);

}

}

**class** manager **extends** employee{

**private** String depart;

**public** manager (String name, **double** salary, String depart) {

**super**(name, salary);

**this**.depart = depart;

}

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

**super**.displayDetails();

System.***out***.println("Department: " + depart);

}

}

**public** **class** news {

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

employee Emp=**new** employee("raj",50000);

manager man=**new** manager("veeraj",500000,"CEO");

System.***out***.println("Empo detail");

Emp.displayDetails();

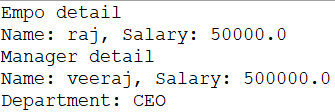
System.***out***.println("Manager detail");

man.displayDetails();

}

}

**OUTPUT**



6.Write the same programme for the class ImmutableExample, to achieve object value ‘Hi’

**CODE**

**package** box;

**public** **final** **class** imu {

**private** String name;

imu (String name) {

**this**.name = name;

}

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

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

imu obj = **new** imu("hi");

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

}

}

**OUTPUT**



7. Write the same programme for the class MutableExample, to output the object values ‘hello 2’ and ‘hello3’.

**CODE**

**package** box;

**public** **class** mu {

**private** String name;

mu(String name) {

**this**.name = name;

}

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

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

mu obj = **new** mu("hello2");

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

obj.setName(" hello3");

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

}

}

**OUTPUT**



8. Write a java class to implement any 10 string methods:

● replace ● contains ● replaceAll ● indexOf ● substring ● Equals ● lastIndexOf ● startsWith

● endsWith ● EqualsIgnoreCase ● toLowerCase ● toUpperCase ● isEmpty ● Length ● split

● replace

**CODE**

**package** box;

**public** **class** str {

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

String str = "veeraj";

System.***out***.println(str.replace('j','i'));

}

}

**OUTPUT**



● indexOf

**CODE**

**package** box;

**public** **class** str {

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

String str = "veeraj";

System.***out***.println(str.indexOf("v"));

}

}

**OUTPUT**



**● toLowerCase**

**package** box;

**public** **class** str {

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

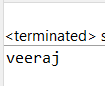
String str = "VEERAJ";

System.***out***.println(str.toLowerCase());

}

}

**OUTPUT**



● toUpperCase

**CODE**

**package** box;

**public** **class** str {

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

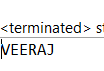
String str = "veeraj";

System.***out***.println(str.toUpperCase());

}

}

**OUTPUT**



● Length

**CODE**

**package** box;

**public** **class** str {

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

String str = "veeraj";

System.***out***.println(str.length());

}

}

**OUTPUT**



**● startsWith**

**CODE**

**package** box;

**public** **class** str {

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

String str = "veeraj";

System.***out***.println(str.startsWith("ve"));

}

}

**OUTPUT**



● endsWith

**CODE**

**package** box;

**public** **class** str {

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

String str = "veeraj";

System.***out***.println(str.endsWith("aj"));

}

}

**OUTPUT**



● EqualsIgnoreCase

**CODE**

**package** box;

**public** **class** str {

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

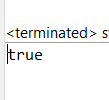
String str = "veeraj";

System.***out***.println(str.equalsIgnoreCase("veeraj"));

}

}

**OUTPUT**



**● split**

**package** box;

**public** **class** str {

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

String str = "veeraj";

String[] strArray = str.split("r");

**for**(String eachString:strArray){

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

}

}

}

**OUTPUT**

