**Lab-05**

**Overloading & Access Control**

**Objectives:**

Understanding concepts method and constructor overloading. Learn how to provide different access controls on class members

**Theory:**

* **Method Overloading**If a class has multiple methods by same name but different parameters, it is known as Method Overloading.

Three ways to overload a method

In order to overload a method, the parameter list of the methods must differ in either of these:

**1. Number of parameters.**

For example: This is a valid case of overloading

add(int, int)

add(int, int, int)

**2. Data type of parameters.**

For example:

add(int, int)

add(int, float)

**3. Sequence of Data type of parameters.**

For example:

add(int, float)

add(float, int)

**Invalid case of method overloading:**

Parameters list doesn’t mean the return type of the method, for example if two methods have same name, same parameters and have different return type, then this is not a valid method overloading example. This will throw a compilation error.

int add(int, int)

float add(int, int)

**Type Promotion table:**

The data type on the left side can be promoted to any of the data type present at the right side.

byte → short → int → long → double

short → int → long → float → double

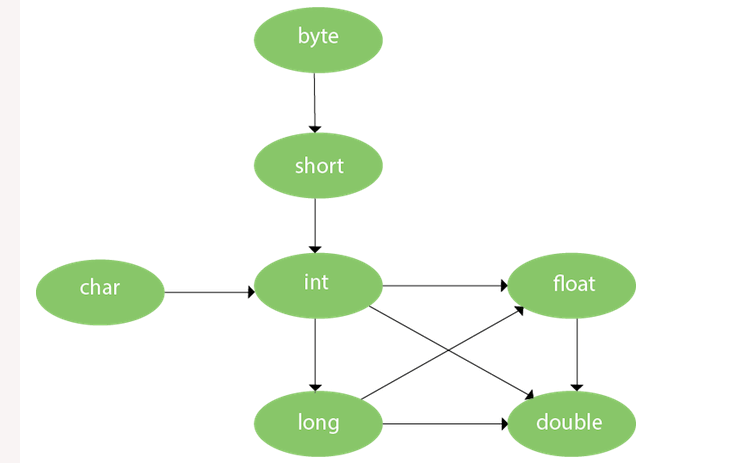
int → long → float → double

float → double

long → float → double

char → int → long → float → double

This can be represented as a diagram like this:



* **Constructor Overloading**If a class has multiple constructors having different parameters, it is known as Constructor Overloading.
* **Access Control**The access modifiers in java specify accessibility (scope) of a data member, method, constructor or class.

There are 4 types of java access modifiers:

1. private
2. default
3. protected
4. public

**Lab Task:**

// Demonstrate method overloading.

classOverloadDemo {

void test() {

System.out.println("No parameters");

}

// Overload test for one integer parameter.

void test(int a) {

System.out.println("a: " + a);

}

// Overload test for two integer parameters.

void test(int a, int b) {

System.out.println("a and b: " + a + " " + b);}

// overload test for a double parameter

double test(double a) {

System.out.println("double a: " + a);

return a\*a; }

OverloadDemo(){

System.out.println("No-args constructor ");}

OverloadDemo(int demo){

System.out.println("Parameterized Constructor :" + demo) ;} }

// \_\_\_\_\_\_\_\_\_\_\_ Calling Class \_\_\_\_\_\_\_\_\_\_\_

class Overload {

public static void main(String args[]) {

OverloadDemo ob = new OverloadDemo();

OverloadDemo ob1 = new OverloadDemo(33);

double result;

// call all versions of test()

ob.test();

ob.test(10);

ob.test(10, 20);

result = ob.test(123.25);

System.out.println("Result of ob.test(123.25): " + result);

}

}

**Lab Assignment:**

**Design a class named Account that contains:**

* A private int data field named id for the account (default 0).
* A private double data field named balance for the account (default 0).
* A private double data field named annualInterestRate that stores the current interest rate (default 0). Assume all accounts have the same interest rate.
* A private Date data field named dateCreated that stores the date when the account was created.
* A no-arg constructor that creates a default account.
* A constructor that creates an account with the specified id and initial balance.
* The accessor and mutator methods for id, balance, and annualInterestRate.
* The accessor method for dateCreated.
* A method named getMonthlyInterestRate() that returns the monthly interest rate.
* A method named getMonthlyInterest() that returns the monthly interest.
* A method named withdraws that withdraws a specified amount from the account.
* A method named deposit that deposits a specified amount to the account.

(Hint: Monthly interest is balance \* monthlyInterestRate.

monthlyInterestRate is annualInterestRate / 12.

Note that annualInterestRate is a percentage. You need to divide it by 100.

Design a test program that creates an Account object with an account ID of 1122, a balance of $20,000, and an annual interest rate of 4.5%. Use the withdraw method to withdraw $2,500, use the deposit method to deposit $3,000, and print the balance, the monthly interest, and the date when this account was created.

**Conclusion:**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**