JavaClass21 Method Overloading:

Method Overloading:

Allows us to have same name methods or constructor but with different parameters.

Why we should use MethodOverloading?

1) It gives the flexibility to the end user that he use the same method in different ways.

2) It helps us write cleaner code.

What is the syntax to overload a Method?

Create different methods but keep the following points in mind.

1) we can overload a method by just passing different number of parameters.

2) we can overload a method by changing the data type of parameter.

3) we can overload a method by changing the sequence.

package class21;

public class Animal {

String name;

String color;

String breed;

void sleep(){

System.out.println(" Animals usually sleep for 6 hours");

}

void eat(){

System.out.println("Grass and meat");

}

}

class Cat extends Animal{

void sleep(){

System.out.println("I like to sleep for 20 hours");

}

void eat(){

System.out.println("i only like fish");

}

}

class Dog extends Animal{

void sleep() {

System.out.println(" I like to sleep for 10 hours");

}

void eat(){

System.out.println("Dogs like meat");

}

}

package class21;

public class AnimalTester {

public static void main(String[] args) {

Cat cat=new Cat();

cat.eat();

Dog dog=new Dog();

dog.eat();

}

}

package class21;

public class Browser {

void openBrowser(){

System.out.println("Opening a Browser");

}

void navigate(){

System.out.println("opening the url");

}

void test(){

System.out.println("Testing the webpage");

}

void closeBrowser(){

System.out.println("closing the browser");

}

}

class Chrome extends Browser{

@Override

public void openBrowser(){

System.out.println("opening the browser in less than 1 sec");

}

}

class FireFox extends Browser{

void openBrowser(){

System.out.println("opening the browser in less than 2 sec");

}

}

package class21;

public class BrowserTester {

public static void main(String[] args) {

Chrome chrome=new Chrome();

chrome.openBrowser();

chrome.closeBrowser();

}

}

package class21;

public class Employee {

String name;

double baseSalary=40000;

double bonus=20000;

void calculateSalary(){

System.out.println(baseSalary+bonus);

}

}

class Developer extends Employee{

void calculateSalary(){

System.out.println(2\*baseSalary+(1.5\*bonus));

}

}

class Tester extends Employee{

void calculateSalary(){

System.out.println(baseSalary+(2\*bonus));

}

}

class OfficeBoy extends Employee{

}

package class21;

public class EmployeeTester {

public static void main(String[] args) {

OfficeBoy officeBoy=new OfficeBoy();

officeBoy.calculateSalary();

Tester tester=new Tester();

tester.calculateSalary();

Developer developer=new Developer();

developer.calculateSalary();

}

}

package class21;

public class Parent {

void getMarry(){

System.out.println("You will Marry the girl that we will select for you");

}

}

class Zam extends Parent{

void getMarry() {

System.out.println("I want to marry taylor swift");

}

}

class Test{

public static void main(String[] args) {

Zam zam=new Zam();

zam.getMarry();

}

}

package class21;

public class Task1 {

/\*Create 1 class in which create a methods that will calculate the area (volume in case of box) of

Rectangle, Square, Box

Use separate class to test your code

\*/

public static void main(String[] args) {

AreaManager areaManager=new AreaManager();

areaManager.calculateArea(10);

areaManager.calculateArea(10,12);

}

}

class AreaManager{

void calculateArea(double len,double wid){

System.out.println(len\*wid);

}

void calculateArea(double len){

System.out.println(len\*len);

}

void calculateVolume(double len,double height, double wid){

System.out.println(len\*wid\*height);

}

}

package class21;

public class Task1C20 {

/\*Write program: userClass that has a constructor that initializes instance variable name and mobile number.

Create a subclass userInfo that will have user address variable and it also being initialized through constructor call.

Print users name, mobile number and address in userDetails method. Test your code.\*/

public static void main(String[] args) {

UserInfo userInfo=new UserInfo("ZamanUllah","+5715322934","Pizzahut King street");

userInfo.printUserDetails();

}

}

class UserClass{

String name;

String mobileNumber;

UserClass(String name,String mobileNumber){

this.name=name;

this.mobileNumber=mobileNumber;

}

}

class UserInfo extends UserClass{

String address;

UserInfo(String name,String mobileNumber,String address){

super(name,mobileNumber);

this.address=address;

}

void printUserDetails(){

System.out.println("Name "+name+" Mobile Number "+mobileNumber+" Address "+address);

}

}

package class21;

public class Task1C20 {

/\*Write program: userClass that has a constructor that initializes instance variable name and mobile number.

Create a subclass userInfo that will have user address variable and it also being initialized through constructor call.

Print users name, mobile number and address in userDetails method. Test your code.\*/

public static void main(String[] args) {

UserInfo userInfo=new UserInfo("ZamanUllah","+5715322934","Pizzahut King street");

userInfo.printUserDetails();

}

}

class UserClass{

String name;

String mobileNumber;

UserClass(String name,String mobileNumber){

this.name=name;

this.mobileNumber=mobileNumber;

}

}

class UserInfo extends UserClass{

String address;

UserInfo(String name,String mobileNumber,String address){

super(name,mobileNumber);

this.address=address;

}

void printUserDetails(){

System.out.println("Name "+name+" Mobile Number "+mobileNumber+" Address "+address);

}

}

package class21;

public class Task2 {

public static void main(String[] args) {

new Programming("Java");

}

/\*

Create a class named 'Programming'.

While creating an object of the class,

if nothing is passed to it,

then the message "I love programming languages" should be printed.

If some String is passed to it,

then in place of "programming languages" the value variable should be printed.

Example, if we pass "Java", then "I love Java" should be printed.

\*/

}

class Programming{

Programming(){

System.out.println("I love programming languages");

}

Programming(String str){

System.out.println("I love "+str);

}

}

package class21;

public class Task2C20 {

/\*

Write program: Shape class has a constructor that takes the radius and has a subclass as circle class.

In circle class create a method to calculate the area of circle. Test your code

\*/

public static void main(String[] args) {

}

}

class Shape{

double radius;

Shape(double radius){

this.radius=radius;

}

public static void main(String[] args) {

Circle circle=new Circle(6);

circle.calculateArea();

}

}

class Circle extends Shape{

Circle(double radius){

super(radius);

}

void calculateArea(){

// System.out.println(3.14\*radius\*radius);

System.out.println(Math.PI\*Math.pow(radius,2));

}

}

package class21;

public class Task3 {

/\*

Create 1 class with a static method that has 3 overloaded forms.

Then call each overloaded method with specific arguments and observe result.

\*/

static void print(){

System.out.println("I love java");

}

static void print(String str){

System.out.println("I love "+str);

}

static void print(String str,int times){

for (int i = 0; i < times; i++) {

System.out.println("I love "+str);

}

}

public static void main(String[] args) {

print("Javascript",5);

}

}

package class21;

public class Task4 {

/\*

Create 1 class with a private method that has 3 overloaded forms.

Then call each overloaded method with specific arguments and observe result.

\*/

private static void print(){

System.out.println("I love java");

}

private static void print(String str){

System.out.println("I love "+str);

}

private static void print(String str,int times){

for (int i = 0; i < times; i++) {

System.out.println("I love "+str);

}

}

public static void main(String[] args) {

print();

}

//Break till 8:33

}