# ACTIVTY 1:

1. In the first part there is not constructor for class Test1 while it’s being instantiated with an input

## Corrected:

public class LAB {

int a;

public static void main(String[] args)

{

LAB t1 = new LAB(5);

}

LAB(int input){

input=a;

}

}

1. There is no method as x() while it’s called

## Corrected:

public class LAB {

public static void main(String[] args)

{

LAB t2 = new LAB();

t2.x();

}

public void x(){

System.out.println("YOu called correct function!!!");

}

}

1. ISSUES:
   1. Class missing
   2. Class not instantiated while method is called
   3. “Public” instead of “public”

## Corrected:

public class LAB {

public static void main(String[] args)

{

Circle c;

c = new Circle();

System.out.println("Radius is " + c.getRadius());

}

}

class Circle{

int radius=45;

public int getRadius(){

return radius;

}

}

1. ISSUES:
   1. Constructor missing
   2. Different data type in class: int in class and double as input argument
   3. Class not instantiated while method is called
   4. Wrong “” used

## Corrected:

public class LAB

{

public static void main(String[] args)

{

C c = new C(5.0);

System.out.println(c.value);

}

}

class C

{

double value = 2;

C(double a){

value=a;

}

}

1. ISSUES:
   1. Private data member accessed outside from the class…

## Corrected:

class Circle

{

public double ra47dius = 1.0;

/\*\* Find the area of this circle \*/

public double getArea()

{

return radius \* radius \* Math.PI;

}

}

public class LAB

{

public static void main(String[] args)

{

Circle myCircle = new Circle();

System.out.println("Radius is "+ myCircle.radius);

}

}

# Activity 2:

NO WE CAN’T CALL A INSTANCE FROM STATIC CONTEXT…

I changed every thing to static and now it’s working….

public class LAB

{

public static void main(String[] args)

{

method1();

}

public static void method1()

{

method2();

}

public static void method2()

{

System.out.println("What is radius " + c.getRadius());

}

static Circle c = new Circle();

}

class Circle{

int radius=45;

public int getRadius(){

return radius;

}

}

# ACTIVITY 3:

ANSWER:

Passing a primitive data type it’s copied while in case of an object its reference is passed

In the following example time is passed by value that’s why it’s not changed while the reference to count (object) is passed and it’s changed even inside the function increment

(primitive data type include int, double and strings etc)

## Code:

class Count

{

public int count;

public Count(int c)

{

count = c;

}

public Count()

{

count = 1;

}

}

public class LAB

{

public static void main(String[] args)

{

Count myCount = new Count();

int times = 0;

for (int i = 0; i < 100; i++)

increment(myCount, times);

System.out.println("count is " + myCount.count);

System.out.println("times is " + times);

}

public static void increment(Count c, int times)//reference of object --- copy(input by value)

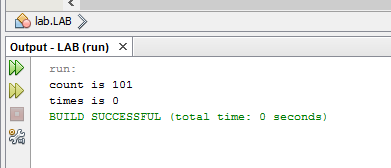
{

c.count++;

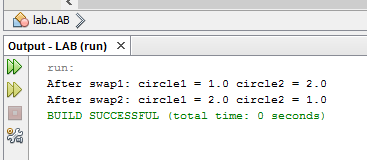
times++;

}

}



# ACTIVITY 4:

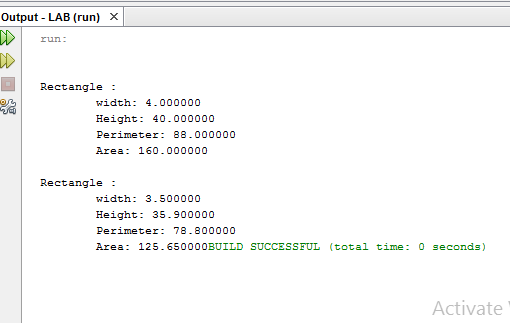


//NO CODE WAS REQUIRED HERE…

# TASK 1:

## UML:

|  |  |  |  |
| --- | --- | --- | --- |
| |  | | --- | | Rectangle | | Width:double  Height:double | | Rectangle()  Rectangle(newWidth:Double,newHeight:Double)  getPerimeter():Double  getArea():Double | |
| |  | | --- | | Rectangle1 | | Width=4  Height=40 |  |  | | --- | | Regtangle2 | | Width=3.5  Height=35.9 | |



## Code:

class Rectangle{

public double width, height;

Rectangle(){

width=1;

height=1;

}

Rectangle(double iwidth,double iheight){

width=iwidth;

height=iheight;

}

double getArea(){

return width\*height;

}

double getPerimeter(){

return 2\*(width+height);

}

}

## Main:

public class LAB

{

public static void main(String[] args){

Rectangle first= new Rectangle(4,40);

displayInfo(first);

Rectangle second= new Rectangle(3.5,35.9);

displayInfo(second);

}

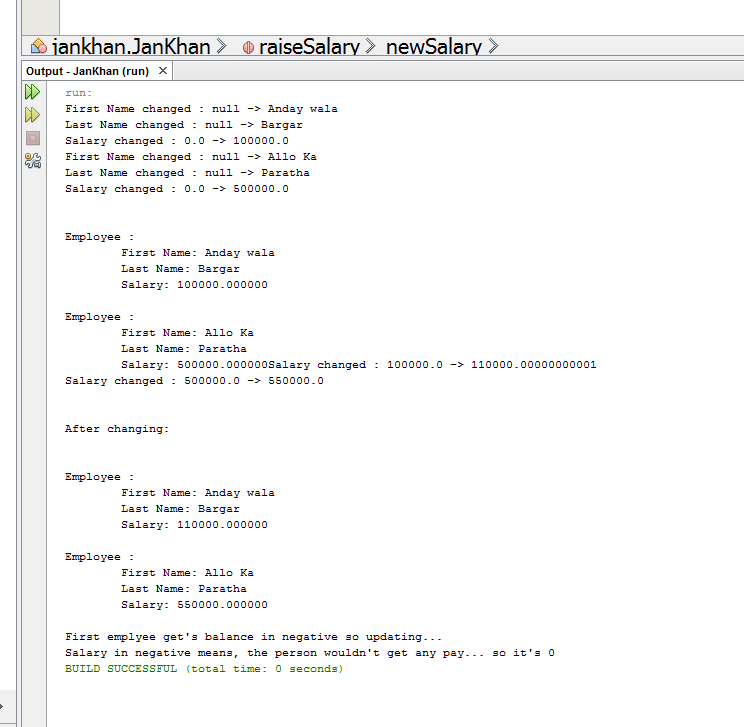
public static void displayInfo(Rectangle r){

System.out.printf("\n\nRectangle : \n\twidth: %f\n\tHeight: %f\n\tPerimeter: %f\n\tArea: %f", r.width,r.height,r.getPerimeter(),r.getArea());

}

}

# Task 02:



## Main:

package jankhan;

import java.util.Random;

import java.util.Scanner;

import jankhan.Employee;

/\*\*

\*

\* @author arifu

\*/

public class JanKhan

{

public static void main(String[] args){

Employee emp1 = new Employee("Anday wala","Bargar",100000);

Employee emp2 = new Employee("Allo Ka","Paratha",500000);

displayInfo(emp1);

displayInfo(emp2);

raiseSalary(emp1,1.1);//1.1 means 10% increase...

raiseSalary(emp2,1.1);

System.out.println("\n\nAfter changing:");

displayInfo(emp1);

displayInfo(emp2);

System.out.println("\n\nFirst emplyee get's balance in negative so updating...");

emp1.setSalary(-1000);

}

public static void displayInfo(Employee emp){

System.out.printf("\n\nEmployee : \n\tFirst Name: %s\n\tLast Name: %s\n\tSalary: %f",emp.getFirstName(),emp.getLastName(),emp.getSalary());

}

public static void raiseSalary(Employee emp, double percentage){

double newSalary = emp.getSalary()\*percentage;

emp.setSalary(newSalary);

}

}

## Employee Class

package jankhan;

/\*\*

\*

\* @author arifu

\*/

public class Employee {

private String firstName;

private String lastName;

private double salary;

public Employee(String first,String last,double isalary){

setFirstName(first);

setLastName(last);

setSalary(isalary);

}

public String getFirstName(){

return firstName;

}

public String getLastName(){

return lastName;

}

public double getSalary(){

return salary;

}

public void setFirstName(String str){

System.out.println("First Name changed : "+ firstName +" -> "+str);

firstName=str;

}

public void setLastName(String str){

System.out.println("Last Name changed : "+ lastName +" -> "+str);

lastName=str;

}

public void setSalary(double newSalary){

if(newSalary>0){

System.out.println("Salary changed : "+ salary +" -> "+newSalary);

salary=newSalary;

}

else{

System.out.println("Salary in negative means, the person wouldn't get any pay... so it's 0");

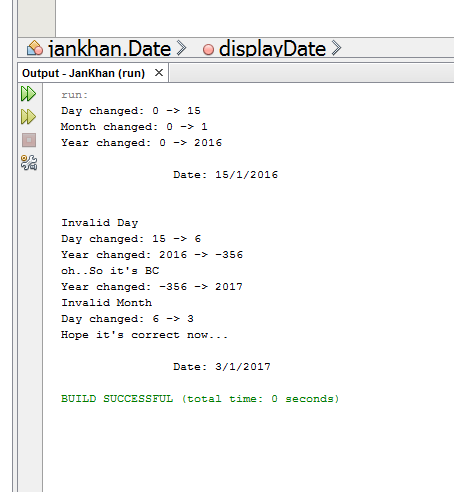
salary=0;

}

}

}

# Task 03:



## MAIN:

package jankhan;

import java.util.Random;

import java.util.Scanner;

import jankhan.Date;

/\*\*

\*

\* @author arifu

\*/

public class JanKhan

{

public static void main(String[] args){

Date today = new Date(1,15,2016);//2117

today.displayDate();

//oh it's wrong.....

today.setDay(33);

today.setDay(6);

today.setYear(-356);

today.setYear(2017);

today.setMonth(13);

today.setDay(3);

System.out.println("Hope it's correct now...");

today.displayDate();

}

}

## DATE.JAVA

package jankhan;

/\*\*

\*

\* @author arifu

\*/

public class Date {

int dmonth;

int dday;

int dyear;

public Date(int month,int day, int year){

setDay(day);

setMonth(month);

setYear(year);

}

public int getDay(){

return dday;

}

public int getMonth(){

return dmonth;

}

public int getYear(){

return dyear;

}

public void setDay(int a){

if(a>0 && a<32){

System.out.println("Day changed: "+dday+ " -> "+a);

dday=a;

}

else{

System.out.println("\nInvalid Day");

}

}

public void setMonth(int a){

if(a>0 && a<13){

System.out.println("Month changed: "+dmonth+ " -> "+a);

dmonth=a;

}

else{

System.out.println("Invalid Month");

}

}

public void setYear(int a){

System.out.println("Year changed: "+dyear+ " -> "+a);

dyear=a;

if(a<0){

System.out.println("oh..So it's BC");

}

}

public void displayDate(){

System.out.printf("\n\t\tDate: %d/%d/%d\n\n",dday,dmonth,dyear);

}

}