|  |  |
| --- | --- |
| **Name** | **M.Asad ur Rehman** |
| **Registration no.** | **2019EE389** |
| **Group** | **G1** |

**Programming Fundamentals**

**Lab # 8 (Class Writing)**

**Objective:**

In this Lab, I understand the concept of class in java.

**Task # 01:**

Design and implement a class called Dog that contains instance data that represents the dog’s name and age. Define the Dog constructor to accept and initialize instance data. Include getter and setter methods for the name and age. Include a method to compute and return the age of the dog in “person years” (seven times the dogs age). Include a toString method that returns a one-line description of the dog. Create a driver class called Kennel, whose main method instantiates and updates several Dog objects.

**Code:**

public class Kennel {

public static void main(String[] args){

Dog x1 = new Dog();

x1.setName("Jar");

x1.setAge(3);

x1.newage();

System.out.println("Data of first Dog :");

System.out.println(x1.toString());

Dog x2 = new Dog();

x2.setName("Hy");

x2.setAge(4);

x2.newage();

System.out.println("Data of second Dog :");

System.out.println(x2.toString());

}

}

public class Dog {

private String name ;

privateint age;

privateintnewage;

public Dog(){}

public void setName(String newname){

name = newname;}

public String getName(){

return name ;}

public void setAge(intnewage){

age = newage;}

publicintgetAge(){

return age;}

public void newage(){

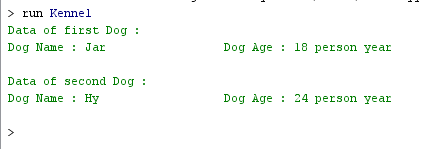
newage = 6\*age;}

public String toString(){

return "Dog Name : "+name+"\t\tDog Age : "+newage+" person year\n";}

}

**Output:**



**Task # 07:**

Design and implement a class called Box that contains instance data that represents the height, width, and depth of the box. Also include a boolean variable called full as instance data that represents whether the box is full or not. Define the Box constructor to accept and initialize the height, width, and depth of the box. Each newly created Box is empty (the constructor should initialize full to false). Include getter and setter methods for all instance data. Include a toString method that returns a oneline description of the box. Create a driver class called BoxTest, whose main method instantiates and updates several Box objects.

**Code:**

public class BoxTest {

public static void main(String[] args){

Box x1= new Box();

x1.setHeight(5);

x1.setWidth(2);

x1.setDepth(6);

System.out.println("Data of first Box :");

System.out.println(x1.toString());

}

}

public class Box {

privateint height;

privateint width;

privateint depth;

public Box(){}

public void setHeight(intnewheight){

height = newheight;}

publicintgetHeight(){

return height;}

public void setWidth(intnewwidth){

width = newwidth;}

publicintgetWidth(){

return width;}

public void setDepth(intnewdepth){

depth = newdepth;}

publicintgetDepth(){

return depth;}

public String toString(){

return "Height : "+height+"cm\t Width : "+width+"cm\t Depth : "+depth+"cm\n";}

}

**Output:**



**Task #03:**

Design and implement a class called Book that contains instance data for the title, author, publisher, and copyright date. Define the Book constructor to accept and initialize this data. Include setter and getter methods for all instance data. Include a toString method that returns a nicely formatted, multi-line description of the book. Create a driver class called Bookshelf, whose main method instantiates and updates several Book objects.

**Code:**

public class BookShelfe {

public static void main(String[] args){

Book x1 = new Book();

x1.setTitle("Never");

x1.setAuth("John");

x1.setPubl("Fleming");

x1.setCopy("Apr");

Book x2 = new Book();

x2.setTitle("Core");

x2.setAuth("Cay ");

x2.setPubl("Prenl");

x2.setCopy("Jun");

System.out.println("Data of first Book : ");

System.out.println(x1.toString());

System.out.println("Data of second Book : ");

System.out.println(x2.toString());

}}

public class Book {

private String title;

private String auth;

private String publ;

private String copy;

Book(){}

voidsetTitle(String newtitle){

title = newtitle;}

String getTitle(){

return title;}

voidsetAuth(String newauth){

auth = newauth;}

String getAuth(){

return auth;}

voidsetPubl(String newpubl){

publ = newpubl;}

String getPubl(){

returnpubl;}

voidsetCopy(String newcopy){

copy = newcopy;}

String getCopy(){

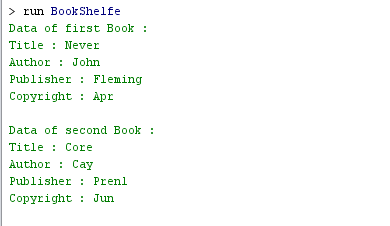
return copy;}

public String toString(){

return "Title : "+title+"\nAuthor : "+auth+"\nPublisher : "+publ+"\nCopyright : "+copy+"\n";}

}

**Output:**



**Conclusion:**

In this lab I perform a task and make a different classes in Java.