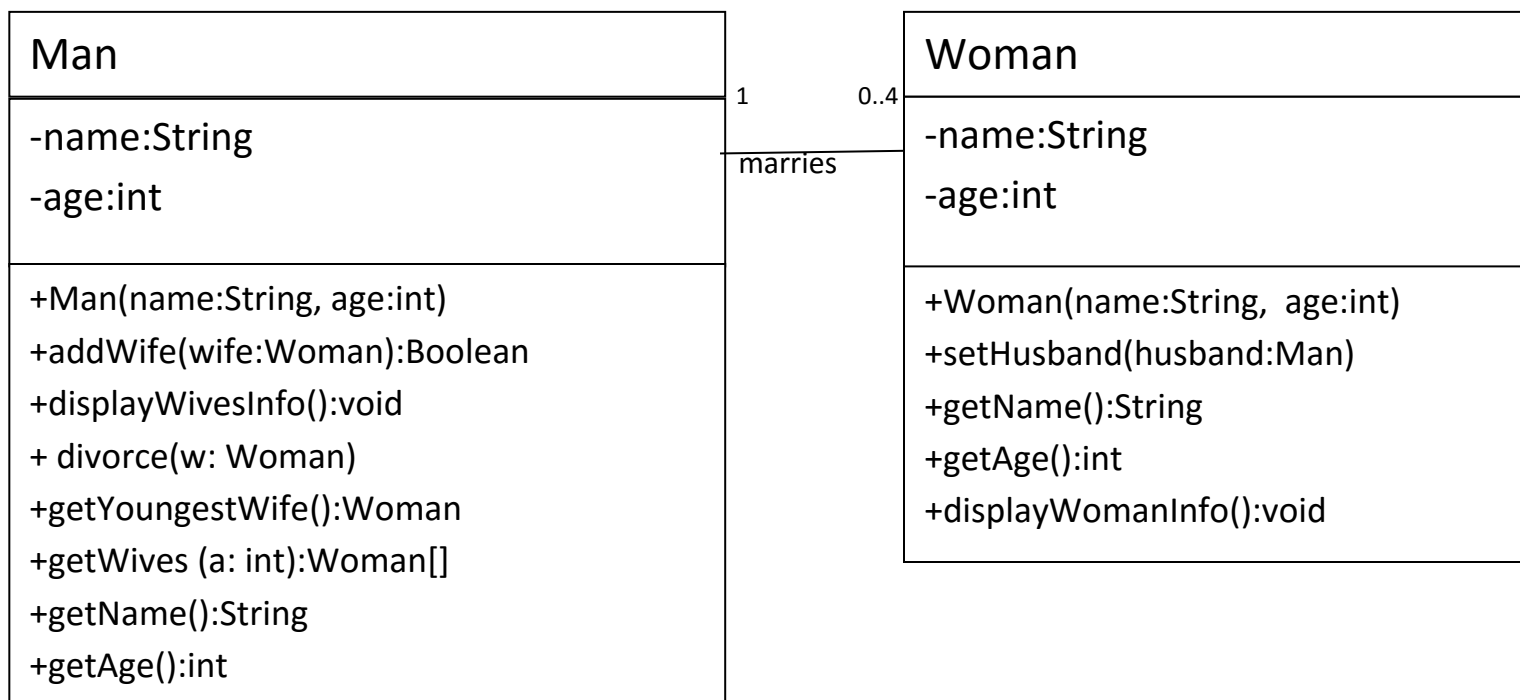


|             |                |                        |
|-------------|----------------|------------------------|
| KSU/CCIS/CS | <b>CSC 113</b> | Tutorial - Association |
|-------------|----------------|------------------------|

Create the classes along with the functionality given in the following UML Diagram. To understand the problem, please refer to the description given after the diagram.



**Woman Class:**

○ Attributes:

- **name**: the name of the Woman.
- **age**: age of the Woman

○ Methods:

- **Woman(name: string, age: int)**: constructor. Assigns given values to the attributes and assigns *null* to *husband*.
- **setHusband(husband:Man)**: sets the husband of the woman.
- **displayWomanInfo()**: this method displays all the attributes of the Woman along with the name and age of Husband if woman is married.
- **getters**: return the values of the corresponding attributes

### ***Man Class:***

#### ○ Attributes:

- ***name***: the name of the Man.
- ***age***: age of the Man

#### ○ Methods:

- ***Man(name:String, age:int)*** : constructor
- ***addWife(wife:Woman)***: this method associates a new wife to the man. It returns true if the ***wife*** is associated false otherwise.
- ***displayWivesInfo()***: displays the detail(name and age) of every wife which is married to the man and in case if man is unmarried , it will display “Ohh..He is unmarried”.
- ***divorce(w: Woman)***: this method removes the Woman w from the wives of the man.
- ***getYoungestWife()***: returns the youngest wife of the man.
- ***getWives(a: int)***: this method returns an array containing all the wives of the man whose age is less than ***a***.
- ***getters***: return the values of the corresponding attributes

### ***HINT: How to Solve Association***

***Man*** class will have an attribute array ***arrwives*** of type ***Woman*** along with a ***counter*** and ***Woman*** class will an attribute ***husband*** of type ***Man***.

```

public class Woman {
    private String name;
    private int age;
    private Man husband;
    public Woman(String name, int age) {
        this.name = name;
        this.age = age;
        husband=null;
    }

    public void setHusband(Man husband) {
        this.husband = husband;
    }

    public String getName() {
        return name;
    }
    public int getAge() {
        return age;
    }

    public void displayWomanInfo()
    {
        System.out.println("Woman Name: "+name);
        System.out.println("Woman Age: "+age);
        if(husband!=null)
        {
            System.out.println("Husband Name:
"+husband.getName());
            System.out.println("Husband Age:
"+husband.getAge());
        }

    }

}

```

```

public class Man {
    private String name;
    private int age;
    private Woman arrWives[];
    int nbWives;
    public Man(String name, int age) {
        this.name = name;
        this.age = age;
        arrWives=new Woman[4];
        nbWives=0;
    }
    public String getName() {
        return name;
    }
    public int getAge() {
        return age;
    }

    public boolean addWife(Woman wife)
    {
        if(nbWives<arrWives.length)
        {
            arrWives[nbWives++]=wife;
            return true;
        }
        else
            return false;
    }

    public void divorce(Woman w)
    {
        boolean found=false;
        for(int i=0;i<nbWives;i++)
            if(arrWives[i].getName().equals(w.getName()))
            {
                arrWives[i]=arrWives[--nbWives];
                System.out.println(w.getName()+" is
divorced");
                found=true;
            }
        if(!found)
            System.out.println("Wife not found");
    }
}

```

```

    public void displayWivesInfo()
    {
        if(nbWives==0)
            System.out.println("Ohh...He is unmarried");
        for(int i=0;i<nbWives;i++)
        {
            System.out.println("Wife "+(i+1)+"
Information:");
            System.out.println("Name:
"+arrWives[i].getName());
            System.out.println("Age: "+arrWives[i].getAge());
        }
    }

    public Woman[] getWives(int a)
    {
        Woman wives[]=new Woman[nbWives];
        int j=0;
        for(int i=0;i<nbWives;i++)

            if(arrWives[i].getAge()<a)
            {
                wives[j]=arrWives[i];
                j++;
            }

        return wives;
    }

    public Woman getYoungestWife()
    {
        Woman youngest=arrWives[0];
        for(int i=1; i<nbWives; i++)
            if(arrWives[i].getAge()<youngest.getAge())
                youngest=arrWives[i];

        return youngest;
    }

}

```