

**COSC 4351 Fall 2023**

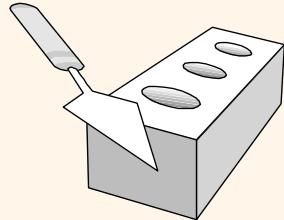
**Software Engineering**

**M & W 4 to 5:30 PM**

Prof. **Victoria Hilford**

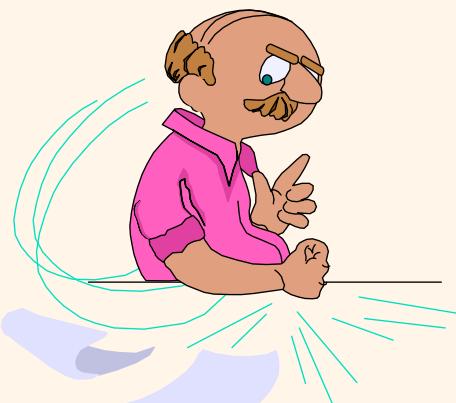
**PLEASE TURN your webcam ON**

**NO CHATTING during LECTURE**



# COSC 4351

## 4 to 5:30



**PLEASE  
LOG IN  
CANVAS**

Youyi [A-L]

Kevin [M-Z]

Please close all other windows.

09.25.2023 (M 4 to 5:30)  (10)	Web App Dev: 1. Guide to Web Application Development 2015.pdf 2. PHP vs C# net Comparison 2015.pdf 3. Ruby on Rails vs PHP Comparison 2015.pdf	Tutorials 2 on Front End Languages C#, RUBY, PYTHON, PHP	OO Languages CANVAS Assignment	
09.27.2023 (W 4 to 5:30)  (11)	Estimating and Planning: 1. Estimating Techniques 2015.pdf 2. How do you estimate on an Agile Project 2015.pdf	Tutorials 3 on C# Visual Studio IDE MVC, PYTHON PyCharm IDE Django, PHP PhpStorm IDE Zend frameworks MECANVAS APPS	WEB APP Papers Summary (1 Page) CANVAS Assignment	
10.02.2023 (M 4 to 5:30)  (12)		Lecture 4: Estimating and Planning Tutorials 4 COCOMO and MS Project	Estimating and Planning Papers Summary (1 Page) CANVAS Assignment	
10.04.2023 (W 4 to 5:30)  (13)		EXAM 2 REVIEW (CANVAS) (ZyBook)	Download ZyBook: Sections 5-9	Q & A Set 2 topics:
10.09.2023 (M 4 to 5:30)  Optional (14)				
10.11.2023 (W 4 to 5:30)  (15)				EXAM 2 (CANVAS) (ZyBook)

# Class 10

## Tutorial 2 on UML to C#, PYTHON, RUBY, PHP5

09.25.2023 (M 4 to 5:30)  (10)	Web App Dev:  1. Guide to Web Application Development 2015.pdf  2. PHP vs ASP.net Comparison 2015.pdf  3. Ruby on Rails vs PHP Comparison 2015.pdf	  Tutorials 2 on Front End Languages C#, RUBY, PYTHON, PHP	  OO Languages  CANVAS Assignment
---	--	---	--

**From 4:00 to 4:10 PM – 10 minutes.**

09.25.2023 (M 4 to 5:30)  (10)	<p>Web App Dev:</p> <p>1. Guide to Web Application Development 2015.pdf</p> <p>2. PHP vs ASP.net Comparison 2015.pdf</p> <p>3. Ruby on Rails vs PHP Comparison 2015.pdf</p>	<p>Tutorials 2 on Front End Languages C#, RUBY, PYTHON, PHP</p>	<p>00 Languages</p> <p>CANVAS Assignment</p>	
---	---	---	--	--

CLASS PARTICIPATION 20 points

20% of Total



## PASSWORD: IN TEAMS



BEGIN Class 10 Participation



CLASS PARTICIPATION 20% Module | Not available until Sep 25 at 4:00pm | Due Sep 25 at 4:10pm | 100 pts

VH – Publish.

09.25.2023  (M 4 to 5:30)  (10)	Web App Dev:  1. Guide to Web Application Development 2015.pdf  2. PHP vs ASP.net Comparison 2015.pdf  3. Ruby on Rails vs PHP Comparison 2015.pdf	<p>Tutorials 2 on Front End Languages C#, RUBY, PYTHON, PHP</p>	00 Languages  CANVAS Assignment
---	--	---	---------------------------------------

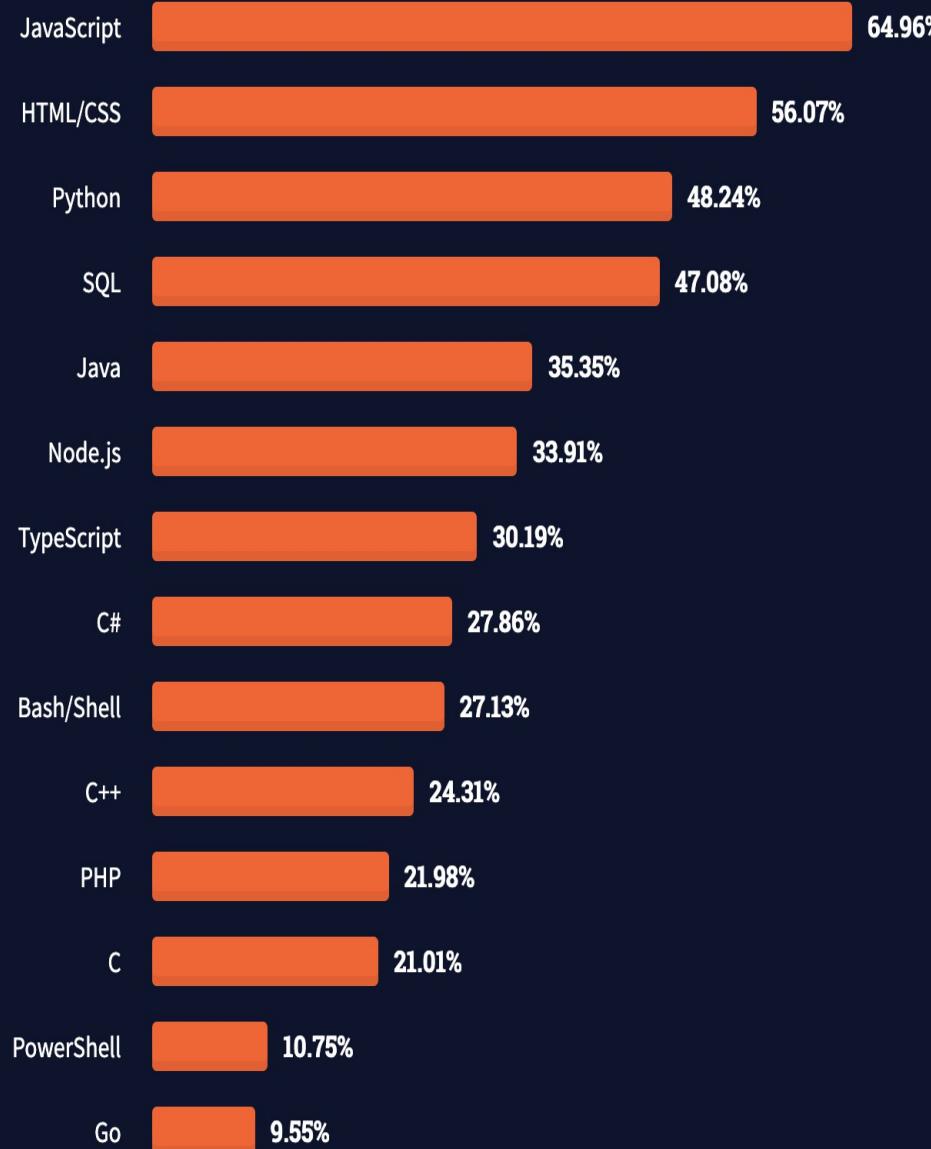
All Respondents

Professional Developers

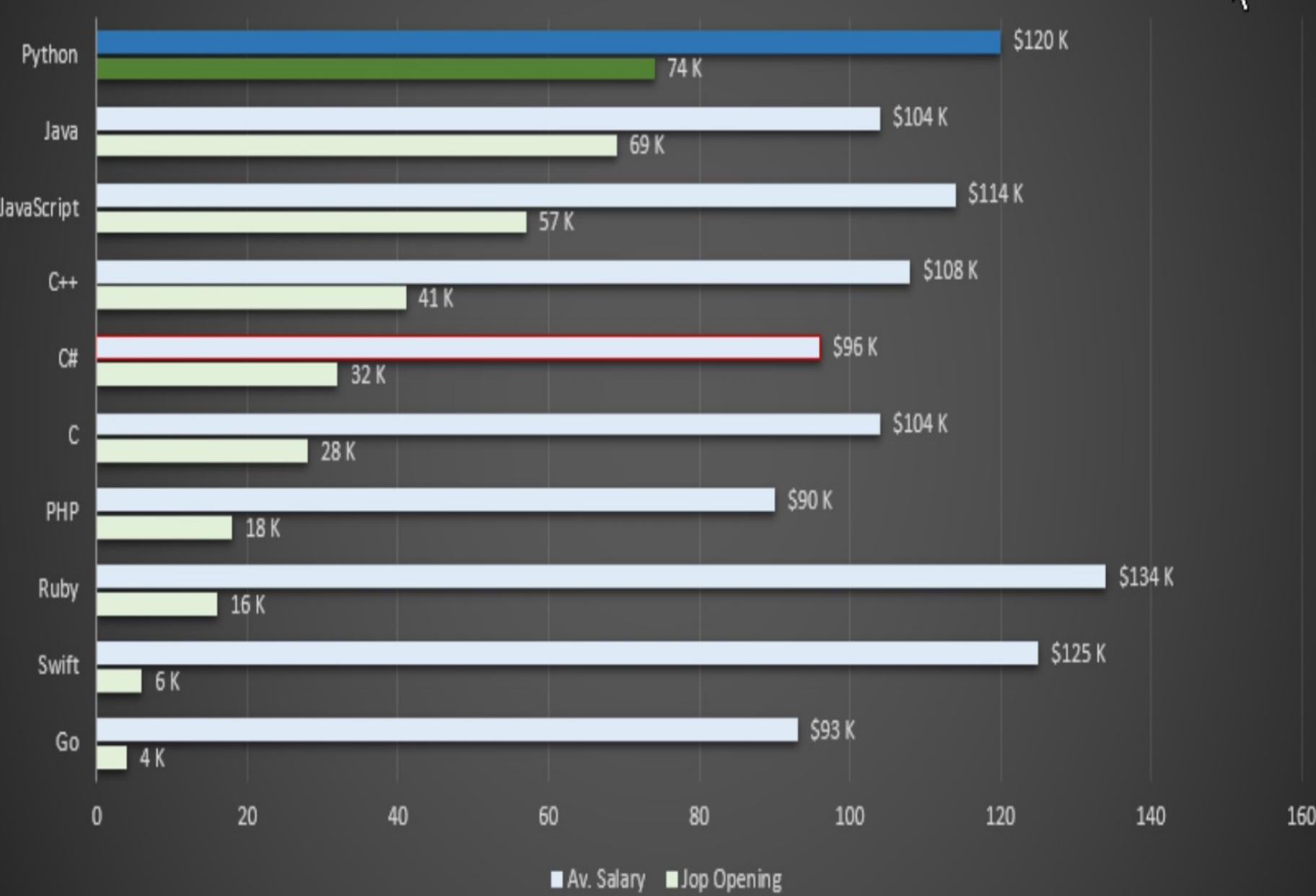
## Top Programming Languages and Web Frameworks To Learn in 2022

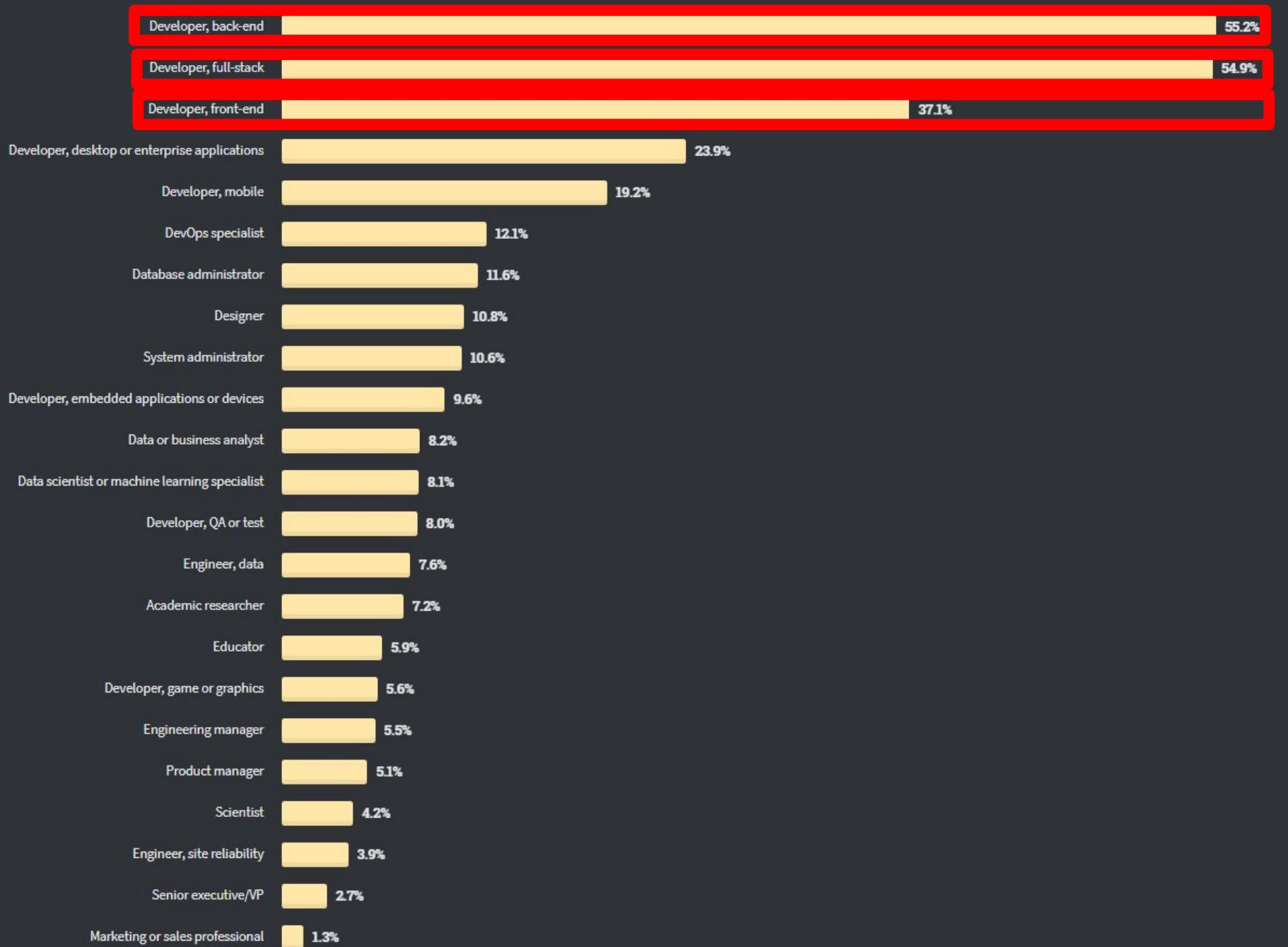
83,052 responses

According to the Stack Overflow survey of over 80,000 developers.



# Av. Salary and Job Openings in USA published in January 2020





# Programming Language Demand Over Time

The table below looks at the demand for various programming languages over the past decade (100 = highest).

YEAR	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019
Swift	8	7	7	7	7	70	59	90	93	83	78
Python	27	27	27	27	30	32	40	52	64	76	90
Java	100	86	80	78	82	77	70	56	46	41	36
JavaScript	99	84	73	70	67	67	73	79	79	77	71
PHP	100	89	75	66	57	51	48	47	40	33	28
SQL	100	93	78	67	58	58	59	54	49	48	46
C++	90	83	71	63	57	52	54	51	46	42	40
Rust	5	5	5	6	8	25	38	43	51	60	86
Elixir	29	27	40	32	33	33	46	71	75	70	70
Scala	42	49	57	47	52	61	69	87	90	80	87
R	53	55	67	81	77	80	82	88	88	87	82
C#	98	92	77	76	73	69	68	72	65	59	54
Ruby	84	73	71	65	67	58	58	56	52	44	39

## TUTORIAL 2 on UML to C#, Ruby, Python, PHP5

Build Content ▾

Assessments ▾

Tools ▾

Partner Content ▾



UML



C# and UML to C#



RUBY and UML to RUBY



PYTHON and UML to PYTHON



PHP and UML to PHP

## Circle Application UML Class Model

TUTORIAL 2 on UML to C#, Ruby, Python, PHP5

Build Content Assessments Tools Partner Content

UML

 Circle Application UML Modeling Step by Step Tutorial.docx  
Enabled: Statistics Tracking

UML

## **Circle Application UML Class Model**

**CIRCLE APPLICATION** Requirements are shown below:

The **CIRCLE APPLICATION** describes circles in terms of the origin

which is a point type with x and y coordinates and a radius.

You need to provide the Constructor with 2 parameters for the  
point Class and the Constructor with 3 parameters for the circle  
Class, the set a point's x and y coordinates, to set a circle's  
radius, to get a point's x and y coordinates, to get a circle's  
radius.

The **CIRCLE APPLICATION** must also compute a circle's circumference  
and the area, a print for a point, and print for a circle(x, y,  
radius, circumference and area).

**How many classes?  
Their Attributes?  
Their METHODS?**

## Circle Application UML Class Model

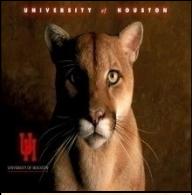
**CIRCLE APPLICATION** Requirements are shown below:

The **CIRCLE APPLICATION** describes circles in terms of the origin which is a point type with x and y coordinates and a radius.

You need to provide the Constructor with 2 parameters for the point Class and the Constructor with 3 parameters for the circle Class, the set a point's x and y coordinates, to set a circle's radius, to get a point's x and y coordinates, to get a circle's radius.

The **CIRCLE APPLICATION** must also compute a circle's circumference and the area, a print for a point, and print for a circle(x, y, radius, circumference and area).

No V or C classes  
JUST a and M  
classes  
**ONE class AT THE  
TIME**



# UML MVC CLASS MODELING

**Domain/ Textual Analysis for**

Second UML MODEL: UML Classes Diagram + Methods Pseudocode



**Step 1: Identify Classes**



**Step 2: Identify Attributes**

Artist 1: name: String  
we a name. The Movie



**Step 3: Identify verbs Methods**

Catalog 1: addArtist(String artist);  
Controller 1: UCAddArtist(String artist);  
pany can add an artist or get



**Step 4: Draw UML Classes Diagram**

**COPY & PASTE**  
**REARANGE**



# CircleApplication

## CircleApplication 1: main ()

### Circle Application UML Class Model

**CIRCLE APPLICATION** Requirements are shown below:

The **CIRCLE APPLICATION** describes circles in terms of the origin

#### Point

#### Point 1: x: double

which is a point type with x and y coordinates and a radius.

#### Point 2: y: double

#### Point 1: Point (x: double,y: double)

You need to provide the Constructor with 2 parameters for the

point Class and the Constructor with 3 parameters for the circle

#### Point 2: setPoint (x: double, y: double): void

Class, the set a point's x and y coordinates, to set a circle's

#### Point 3: getX (): double

radius, to get a point's x and y coordinates, to get a circle's

#### Point 4: getY (): double

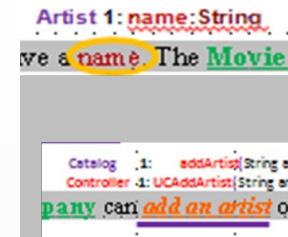
radius.

The **CIRCLE APPLICATION** must also compute a circle's circumference

#### Point 5: print (): double

and the area, a print for a point, and print for a circle(x, y,

radius, circumference and area).



# CircleApplication

## CircleApplication 1: main ()

### Circle Application UML Class Model

**CIRCLE APPLICATION** Requirements are shown below:

#### Circle

The **CIRCLE APPLICATION** describes circles in terms of the origin

which is a point type with x and y coordinates and a radius.

#### Circle 1: radius: double

You need to provide the Constructor with 2 parameters for the

#### Circle 1: Circle (x: double,y: double,r: double)

point Class and the Constructor with 3 parameters for the circle

#### Circle 2: setRadius (r: double): void

Class, the set a point's x and y coordinates, to set a circle's

radius, to get a point's x and y coordinates, to get a circle's

radius.

#### Circle 3: getRadius (): double

#### Circle 4: computeCircumference (): double

The **CIRCLE APPLICATION** must also compute a circle's circumference

#### Circle 5: computeArea (): double

and the area, a print for a point, and print for a circle(x, y,

radius, circumference and area).

#### Circle 6: print (): double

Artist 1: name:String  
we & name. The Movie

Catalog 1: addArtist(String artist)  
Controller 1: UCAddArtist(String artist)  
pany can add an artist

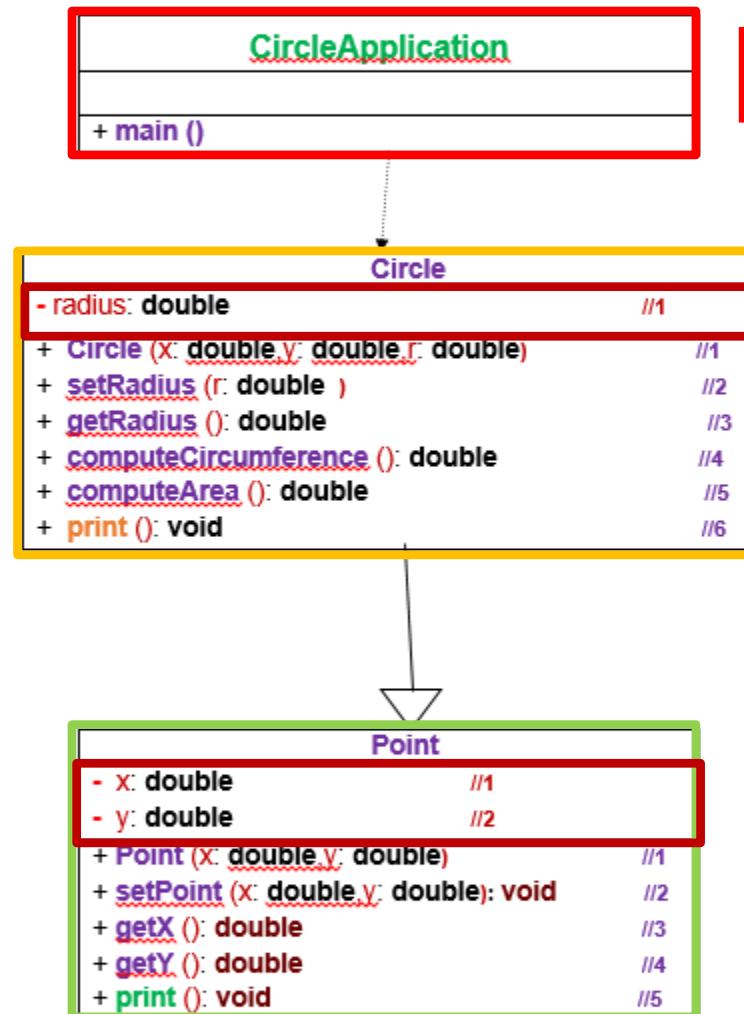
COPY & PASTE  
REARANGE

# Step 4: Draw UML Classes Diagram

UML Class Model:

# Step 4: Draw UML Classes Diagram

UML Class Model:



APPLICATION CLASS

DERIVED CLASS

IS-A inheritance

BASE CLASS

# The DVD Collection System

## UML MVC OOD Class Diagram



Create the instance of the Catalog Class

Catalog catalog = new Catalog();

```

public void UCAddDVD(String title, String category, Double runtime, int year,
Double price) {
    catalog.addDVD(title, category, runtime, year, price);
}
  
```

```

public void addDVD(String title, String category, Double runtime, int year, Double
price) {
    If the title is not already in the catalog
    If(numDVD < MAX_DVD)
        DVD[numDVD] = new DVD(title, category, runtime, year, price);
        numDVD++;
}
  
```

```

public string getTitle() {
    return title;
}
  
```

pseudocode

From 4:20 to 4:30 – 10 minutes.

## Circle Application UML to C#

### TUTORIAL 2 on UML to C#, Ruby, Python, PHP5

Build Content

Assessments

Tools

Partner Content



C# and UML to C#

C#



C# Tutorial.pptx

Enabled: Statistics Tracking



Circle Application UML to C# Step by Step Tutorial.docx

Enabled: Statistics Tracking

# **C# TUTORIAL**

# C# member data

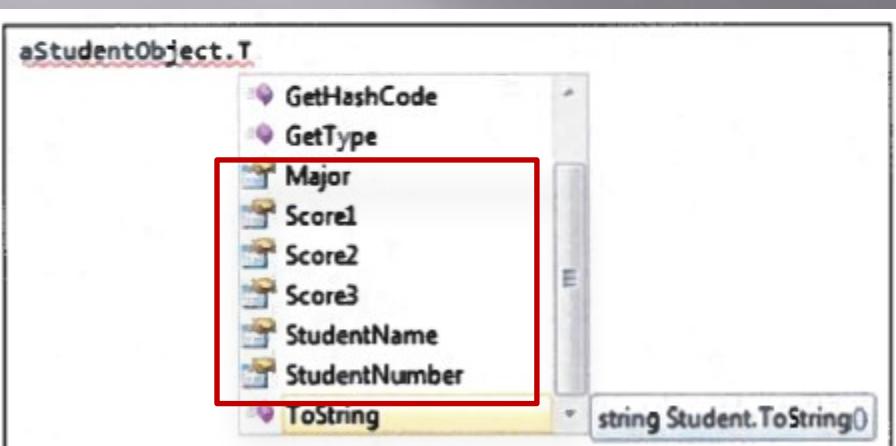


Figure 4-5 Public members of the Student class

```
public class Student
{
    //Data members, data fields, or characteristics
    private int studentNumber;
    private string studentName;
    private int score1;
    private int score2;
    private int score3;
    private string major;
```

# C# member functions

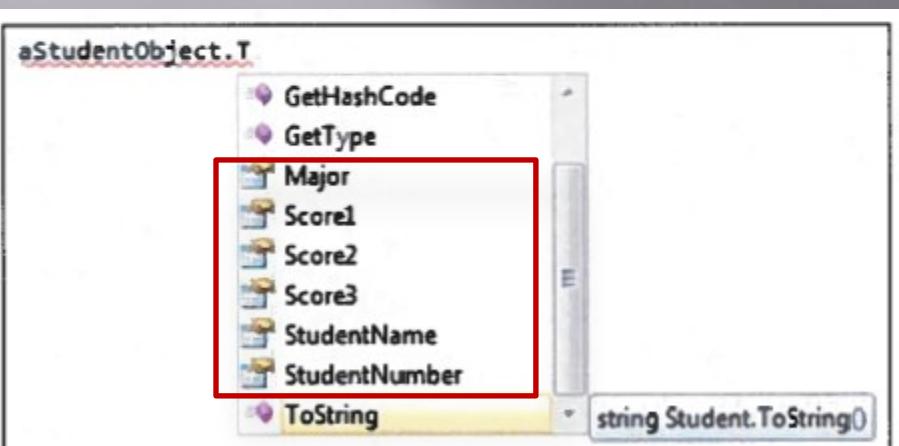


Figure 4-5 Public members of the Student class

```
//default constructor
public Student()
{
}

//constructor with one argument
public Student(int sID)
{
    studentNumber = sID;
}

//constructor with two arguments
public Student(int sID, string name)
{
    studentNumber = sID;
    studentName = name;
}

//constructor with six arguments
public Student(int sID, string name,
              int s1, int s2, int s3, string maj)
{
    studentNumber = sID;
    studentName = name;
    score1 = s1;
    score2 = s2;
    score3 = s3;
    major = maj;
}
```

# C# member functions

aStudentObject.T

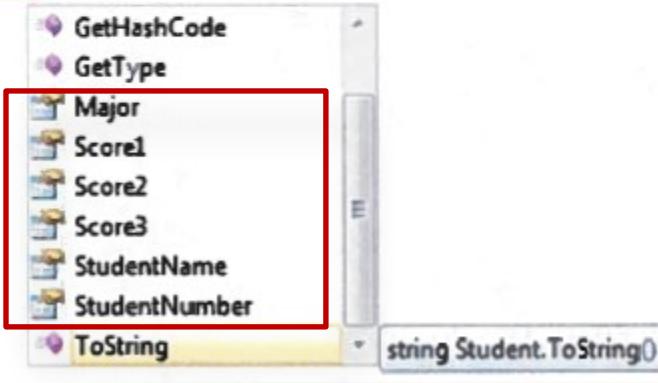


Figure 4-5 Public members of the Student class

```
public double CalculateAverage( )
{
    return (score1 + score2 + score3) / 3.0;
}
```

```
public override string ToString()
{
    return "Name: " +
        studentName +
        "\nMajor: " +
        major +
        "\nScore Average: " +
        CalculateAverage().ToString("f2");
}
```

```
//Properties
public string StudentName
{
    get
    {
        return studentName;
    }
    set
    {
        studentName = value;
    }
}
```

```
public int StudentNumber
{
    get
    {
        return studentNumber;
    }
    set
    {
        studentNumber = value;
    }
}
```

```
public string Major
{
    get
    {
        return major;
    }
    set
    {
        major = value;
    }
}
```

```
public int Score1
{
    get
    {
        return score1;
    }
    set
    {
        score1 = value;
    }
}
```

# Circle Application UML to C#

CSharp - Microsoft Visual Studio

File Edit View Project Build Debug Team Tools Test Analyze Window Help

Quick Launch (Ctrl+Q) Sign in

Point.cs\* Program.cs

Solution Explorer

Search Solution Explorer (Ctrl+Shift+F)

Solution 'CSharp' (1 project)

- CSharp
  - Properties
  - References
  - App.config
  - Point.cs (selected)
  - Program.cs

Server Explorer Toolbox

Point.cs

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace CSharp
8  {
9      class Point
10     {
11         private double x;
12         private double y;
13
14         public Point(double thisX, double thisY)
15         {
16             x = thisX;
17             y = thisY;
18         }
19
20         public void setPoint(double thisX, double thisY)
21         {
22             x = thisX;
23             y = thisY;
24         }
25
26         public double getX()
27         {
28             return x;
29         }
30
31         public double getY()
32         {
33             return y;
34         }
35
36         public void print()
37         {
38             Console.WriteLine("(" + x + ", " + y + ")");
39         }
40     }
41 }
42
43
44
45 }
```

Class Diagram

Point

- X: double //1
- y: double //2
- + Point (x: double,y: double) //1
- + setPoint (x: double,y: double): void //2
- + getX (): double //3
- + getY (): double //4
- + print (): void //5

JAVA?

Solution Explorer Team Explorer Class View

# Circle Application UML to C#

CSharp - Microsoft Visual Studio

Edit View Project Build Debug Team Tools Test Analyze Window Help

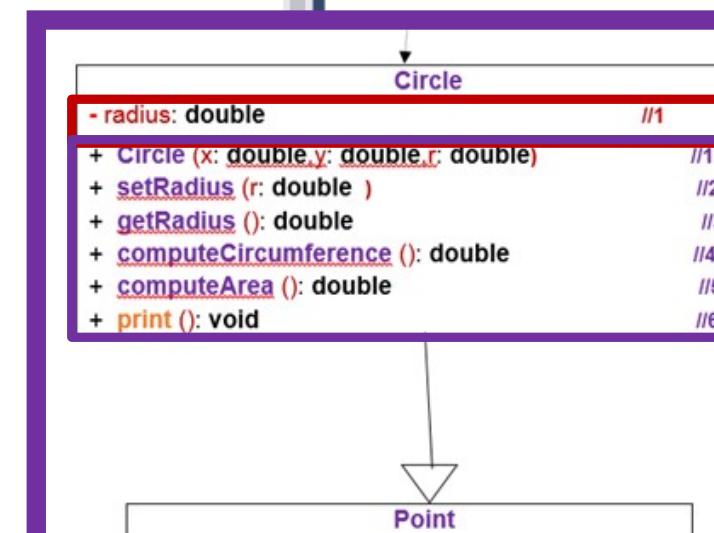
Circle.cs

```
6  
7  namespace CSharp  
8  {  
9      class Circle : Point  
10     {  
11         private double radius;  
12  
13         public Circle(double thisX, double thisY, double thisR) : base(thisX, thisY)  
14         {  
15             radius = thisR;  
16         }  
17  
18         public void setRadius(double thisR)  
19         {  
20             radius = thisR;  
21         }  
22  
23         public double getRadius()  
24         {  
25             return radius;  
26         }  
27  
28         public double getCircumference()  
29         {  
30             return (2 * 3.1416 * radius);  
31         }  
32  
33         public double getArea()  
34         {  
35             return (3.1416 * radius * radius);  
36         }  
37  
38         public void print()  
39         {  
40             Console.WriteLine("Center: ");  
41             base.print();  
42             Console.WriteLine();  
43  
44             Console.WriteLine("Radius: " + radius);  
45             Console.WriteLine("Circumference: " + getCircumference());  
46             Console.WriteLine("Area: " + getArea());  
47         }  
48     }  
49 }
```

Solution Explorer

Search Solution Explorer (Ctrl+.)

Solution 'CSharp' (1 project)  
CSharp  
Properties  
References  
App.config  
Circle.cs  
Point.cs  
Program.cs



JAVA?

Solution Explorer Team Explorer Class View

Ln 9

Col 11

Ch 11

# Circle Application UML to C#

The screenshot shows a Windows desktop environment with several open windows:

- Visual Studio IDE:** The main window displays the code for `Circle.cs` and `CSharp.Program`. The `Circle.cs` file contains UML annotations for class members. The `CSharp.Program` file contains C# code for creating circles, setting their points, and printing their properties.
- Terminal Window:** A terminal window titled "file:///E:/TUTORIALS/COSC4351/TUTORIAL 2 on UML to C# Ruby Python PHP5" shows the output of the application. It prints two instances of the Circle class: "Circle 1" and "Circle 1 modified". Both instances have a radius of 4, a circumference of 25.1328, and an area of 50.2656. The second instance has its center moved to (-2.5, 7).
- Diagram:** A UML class diagram is visible on the right side of the screen. It shows two classes: `Circle` and `Point`. `Circle` has a private attribute `radius: double` and methods: `Circle(x: double, y: double, r: double)`, `setRadius(r: double)`, `getRadius(): double`, `computeCircumference(): double`, `computeArea(): double`, and `print(): void`. `Point` has private attributes `x: double` and `y: double` and methods: `Point(x: double, y: double)`, `setPoint(x: double, y: double): void`, `getX(): double`, `getY(): double`, and `print(): void`.

**Circle.cs (UML Annotated):**

```
class Circle {
    -radius: double //1
    + Circle(x: double, y: double, r: double) //2
    + setRadius(r: double) //3
    + getRadius(): double //4
    + computeCircumference(): double //5
    + computeArea(): double //6
    + print(): void //6
}
```

**CSharp.Program (C# Code):**

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace CSharp
{
    class Program
    {
        static void Main(string[] args)
        {
            Circle circle1 = new Circle(3, 2.5, 4);

            Console.WriteLine("***** Circle 1 *****");
            circle1.print();
            Console.WriteLine();

            circle1.setPoint(-2.5, 7);
            circle1.setRadius(4);

            Console.WriteLine("***** Circle 1 modified *****");
            circle1.print();
            Console.WriteLine();

            double x, y, r;

            Circle circle2 = new Circle(0, 0, 0);

            Console.WriteLine("Enter x Coordinates of the center: ");
            x = Convert.ToInt32(Console.ReadLine());
            Console.WriteLine();

            Console.WriteLine("Enter y Coordinate of the center: ");
            y = Convert.ToInt32(Console.ReadLine());
            Console.WriteLine();

            Console.WriteLine("Enter radius: ");
            r = Convert.ToInt32(Console.ReadLine());
            Console.WriteLine();

            circle2.setPoint(x, y);
            circle2.setRadius(r);

            Console.WriteLine("***** Circle 2 *****");
            circle2.print();
            Console.WriteLine();
        }
    }
}
```

**CircleApplication (Title Bar):**

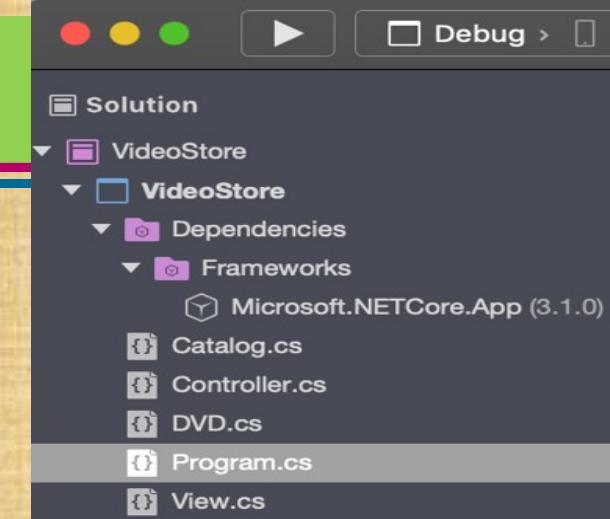
CircleApplication

+ main()

**JAVA?**

# The DVD Collection System

## UML MVC to C#



```

public class DVD
{
    private string title;
    private string category;
    private double run_time;
    private int year;
    private double price;

    public DVD(string thisTitle, string thisCategory, double
thisRunTime,
               int thisYear, double thisPrice)
    {
        title = thisTitle;
        category = thisCategory;
        run_time = thisRunTime;
        year = thisYear;
        price = thisPrice;
    }

    public string getCategory() => category;

    public string getTitle() => title;

    public double getRunTime() => run_time;

    public int getYear() => year;

    public double getPrice() => price;
}

```

```

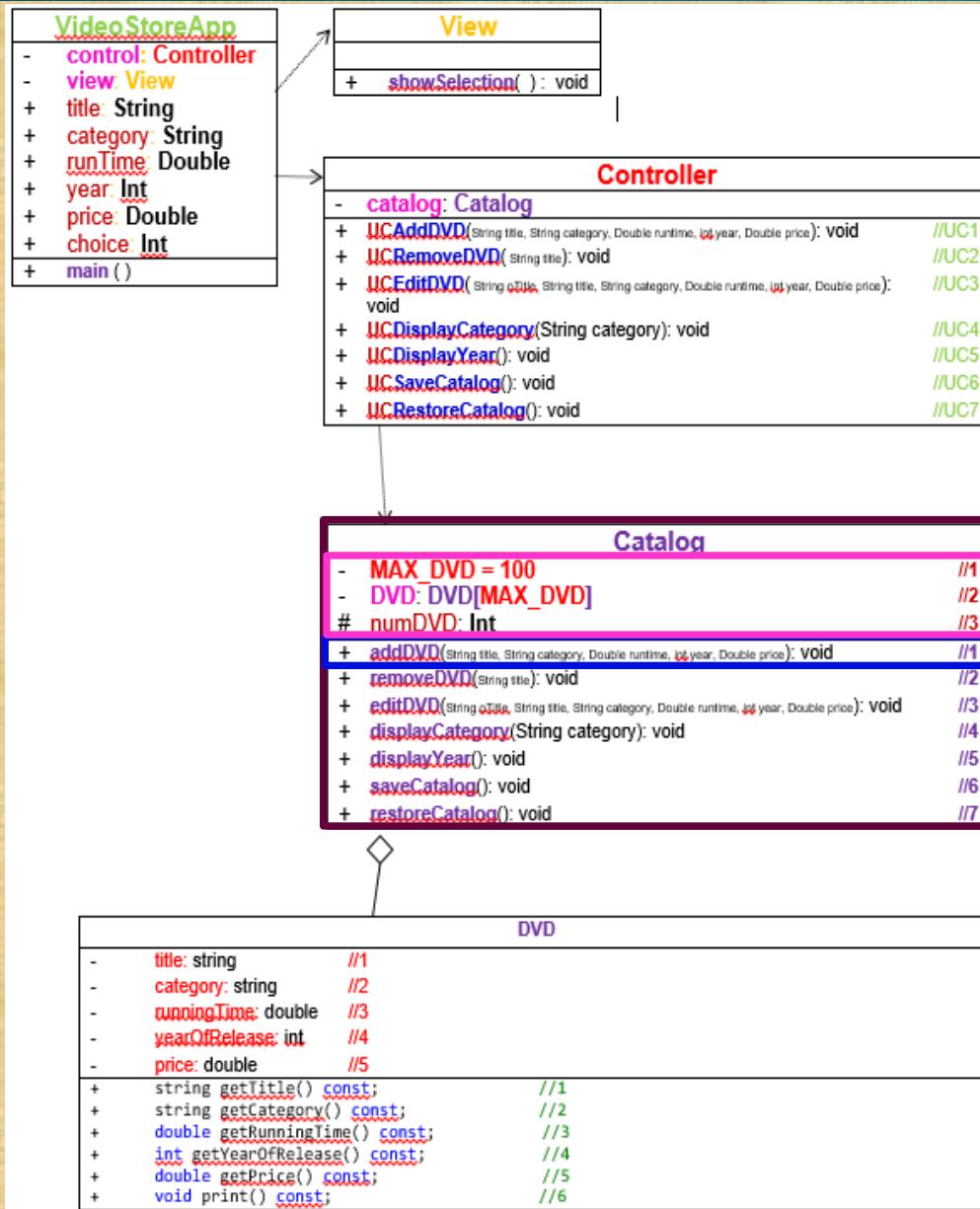
public string getTitle(){
    return title;
}

```

pseudocode to C#

# The DVD Collection System

## UML MVC to C#



```

public class Catalog
{
    private const int MAX_DVD = 100;
    private int numDVD;
    private DVD[] dvd;

    public Catalog()
    {
        numDVD = 0;
        dvd = new DVD[MAX_DVD];
    }

    public void addDVD(string title, string category, double
run_time,
                     int year, double price)
    {
        int exists = 0;

        for (int i = 0; i < numDVD; i++)
        {
            if (dvd[i].getTitle() == title)
            {
                exists = 1;
            }
        }
        if (numDVD < MAX_DVD && exists != 1)
        {
            dvd[numDVD] = new DVD(title, category, run_time, year,
price);
            numDVD++;
        }
    }

    public void addDVD(String title, String category, Double runtime, int year, Double
price)
    {
        If the title is not already in the catalog
        If(numDVD < MAX_DVD)
            DVD[numDVD] = new DVD(title, category, runtime, year, price);
            numDVD++;
    }
}
  
```

pseudocode to C#

# The DVD Collection System

## UML MVC to C#



Create the instance of the Catalog Class

```
Catalog catalog = new Catalog();
```

```
public void UCAddDVD(String title, String category, Double runtime, int year, Double price) {
    catalog.addDVD(title, category, runtime, year, price);
}
```

```
public class Controller
```

```
{
    private Catalog catalog;

    public Controller()
    {
        catalog = new Catalog();
    }
}
```

```
public void UCAddDVD(string thisTitle, string thisCategory,
                     double thisRuntime, int thisYear, double thisPrice)
{
    catalog.addDVD(thisTitle, thisCategory, thisRuntime,
                   thisYear,
                   thisPrice);
}
```

pseudocode to C#

## Circle Application UML to Ruby

### TUTORIAL 2 on UML to C#, Ruby, Python, PHP5

Build Content

Assessments

Tools

Partner Content



RUBY and UML to RUBY



Ruby Tutorial.pptx

Enabled: Statistics Tracking

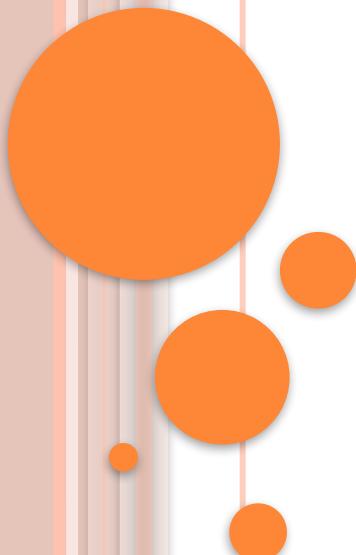


Circle Application UML to Ruby Step by Step Tutorial.docx

Enabled: Statistics Tracking

Ruby

# Ruby Tutorial



## WHAT IS **Ruby**?

- an **interpreted**, general-purpose high-level programming language whose design philosophy emphasizes code readability
- **O**bject Oriented Programming (**OOP**)
- Originated in **Japan in mid-1990's** by **Yukihiro Matsumoto**



# Hello

# World!

JAVA

```
public class Main
{
    public static void main(String[] args)
    {
        System.out.println("Hello World!");
    }
}
```

C++

```
#include <iostream>
using namespace std;

int main()
{
    cout << "Hello World!" << endl;
}
```



# Hello World!

Ruby

puts "Hello World!"





# Ruby!

- Useful as a **scripting language**
  - **script**: A **small program** meant for one time use
  - Targeted towards **small to medium size projects**
- Use by:
  - Amazon, Twitter, Yahoo!, White Pages, Reddit



# Interpreted

---

- **C/C++**
  - **Compiled** to assembly/Run directly on machine
- **Java**
  - **Compiled** to bytecode/**Interpreted** by JVM
- **Ruby**
  - **Interpreted** (no compilation)



# Our First Program

- **Ruby** does not have a **main** method like **Java**
  - Just write your code directly in a file
- **Ruby** statements do not end with **semicolons**
- **Method** calls don't need parenthesis ( )

```
File Edit Options Buffers Tools
def say_hello
    puts "Hello World"
end

# main calls say hello twice
say_hello
say hello
```

# Circle Application UML to Ruby

Ruby - NetBeans IDE 6.9.1

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

<default>

Search (Ctrl+I)

Projects Files Services

Ruby  
Source Files  
Point.rb  
main.rb  
Test Files  
RSpec Files  
Libraries  
Rakefile  
LICENSE  
README

Point

- x: double //1
- y: double //2
- + Point (x: double,y: double) //1
- + setPoint (x: double,y: double): void //2
- + getX (): double //3
- + getY (): double //4
- + print (): void //5

main.rb x Point.rb x

```
1 class Point
2   attr_accessor :x, :y
3
4   def initialize (x, y)
5     @x = x
6     @y = y
7   end
8
9   def setPoint(x, y)
10    @x = x
11    @y = y
12  end
13
14  def getX()
15    return @x
16  end
17
18  def getY()
19    return @y
20  end
21
22  def printPoint()
23    print '[' , @x , ',' , @y , ']'
24    puts
25  end
26
27
28
29
```

28 | 1 INS

# Circle Application UML to Ruby

Ruby - NetBeans IDE 6.9.1

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Projects Files Services

Ruby

- Source Files
  - Circle.rb
  - Point.rb
  - main.rb
- Test Files
- RSpec Files
- Libraries
- Rakefile
- LICENSE
- README

main.rb x Point.rb x Circle.rb x

require 'Point'

class Circle < Point

attr\_accessor :x, :y, :radius

?? Inheritance

```
def initialize(x, y, r)
    super(x, y)
    @radius = r
end

def setRadius(r)
    @radius = r
end

def getRadius()
    return @radius
end

def getCircumference()
    return (2 * 3.1416 * @radius)
end

def getArea()
    return (3.1416 * @radius * @radius)
end

def printCircle()
    print 'Center: '
    printPoint()
    print "Radius: ", @radius
    puts      #adds a new line
    print "Circumference: ", getCircumference()
end
```

Circle

- radius: double //1

+ Circle (x: double, y: double, r: double) //1

+ setRadius (r: double) //2

+ getRadius (): double //3

+ computeCircumference (): double //4

+ computeArea (): double //5

+ print (): void //6

Point

```
graph TD
    Circle --> Point
```

# Circle Application UML to Ruby

Ruby - NetBeans IDE 6.9.1

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Files Services

Ruby

- Source Files
  - Circle.rb
  - Point.rb
  - main.rb
- Test Files
- RSpec Files
- Libraries
- Rakefile
- LICENSE
- README

main.rb Point.rb Circle.rb

```
1 require 'Circle'
2
3 circle1 = Circle.new(3, 2.5, 4)
4 puts "***** Circle 1 *****"
5 circle1.printCircle()
6 puts
7
8 circle1.setPoint(-2.5, 7)
9 circle1.setRadius(4)
10 puts "***** Circle 1 modified *****"
11 circle1.printCircle()
12 puts
13
14 circle2 = Circle.new(0, 0, 0)
15 print 'Enter x Coordinates of the center: '
16 x = gets.chomp.to_i
17 print 'Enter y Coordinates of the center: '
18 y = gets.chomp.to_i
```

Output - Ruby

```
***** Circle 1 *****
Center: [3 , 2.5]
Radius: 4
Circumference: 25.1328
Area: 50.2656

***** Circle 1 modified *****
Center: [-2.5 , 7]
Radius: 4
Circumference: 25.1328
Area: 50.2656

Enter x Coordinates of the center: |
```

Circle

- radius: double //1
- + Circle (x: double, y: double, r: double) //2
- + setRadius(r: double) //3
- + getRadius(): double //4
- + computeCircumference(): double //5
- + computeArea(): double //6
- + print(): void //6

Point

- x: double //1
- y: double //2
- + Point (x: double, y: double) //1
- + setPoint(x: double, y: double): void //2
- + getX(): double //3
- + getY(): double //4
- + print(): void //5

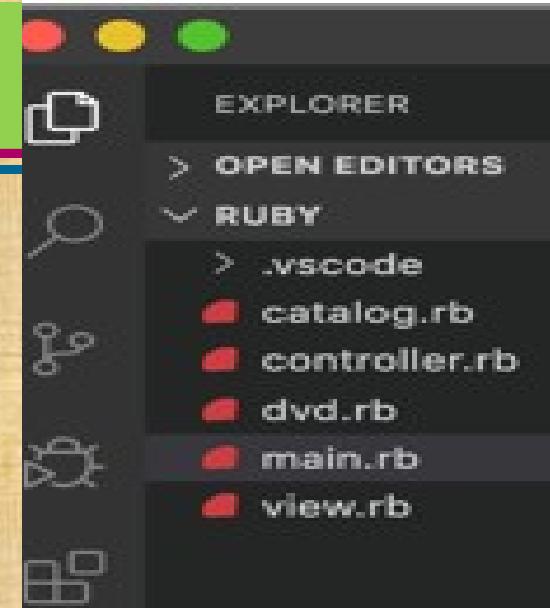
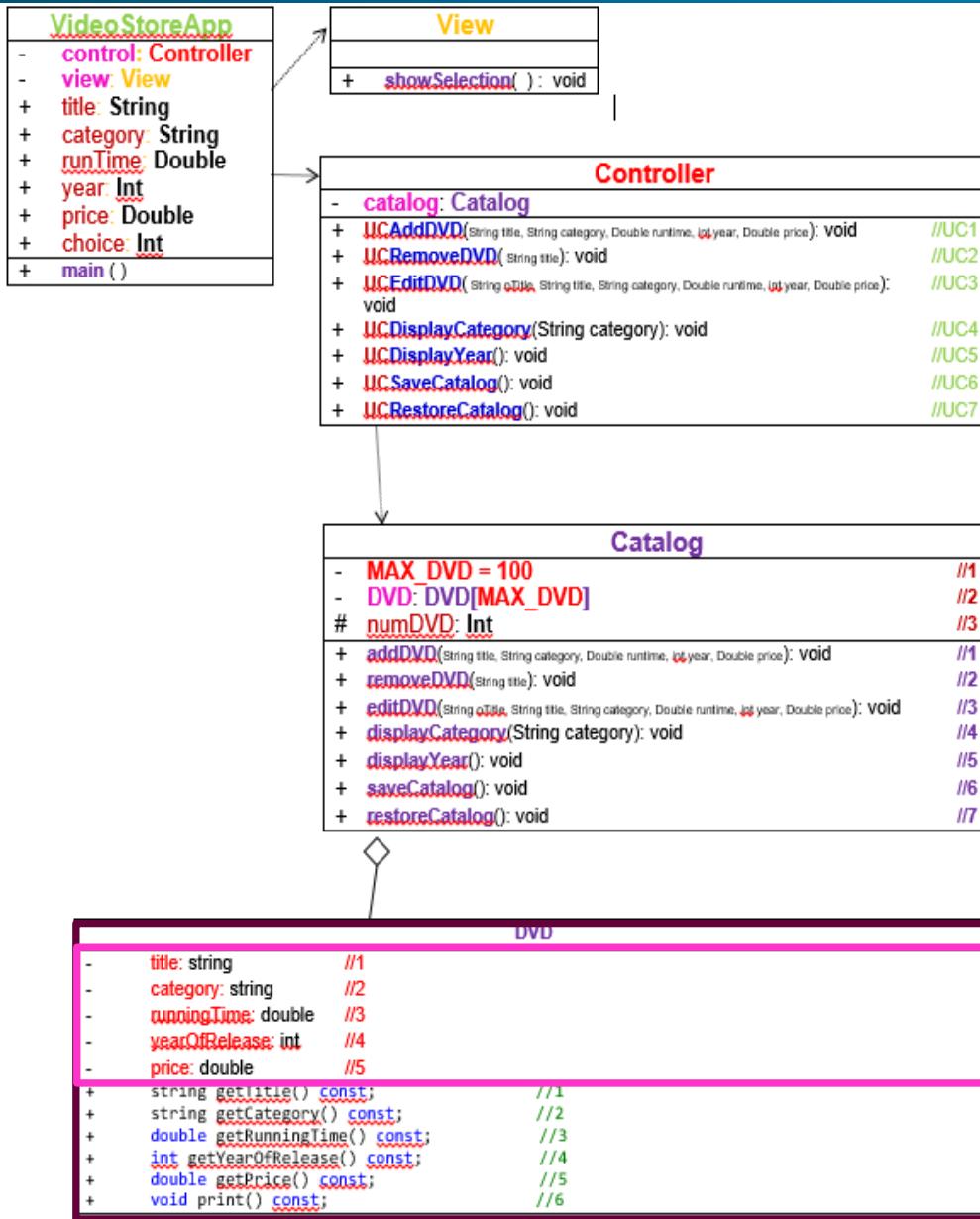
CircleApplication

- + main()

Ruby Running 1 | 16 INS

# The DVD Collection System

## UML MVC to ruby



```

class DVD
  attr_accessor :title, :category, :run_time, :year, :price

  def initialize(title, category, run_time, year, price)
    @title = title
    @category = category
    @run_time = run_time
    @year = year
    @price = price
  end
end
  
```

pseudocode to ruby

# The DVD Collection System

## UML MVC to ruby



```

class Catalog
    MAX_DVD = 100
    @number_dvd = 0
    @dvd = []

```

```

def add_dvd(title, category, run_time, year, price)
    unless dvd.title.include? category
        if number_dvd < MAX_DVD
            @dvd << DVD.new(title, category, run_time, year, price)
            @number_dvd += 1
        end
    end
end

```

```

public void addDVD(String title, String category, Double runtime, int year, Double price) {
    If the title is not already in the catalog
    If(numDVD < MAX_DVD)
        DVD[numDVD] = new DVD(title, category, runtime, year, price);
        numDVD++;
}

```

pseudocode to ruby

# The DVD Collection System

## UML MVC to ruby



Create the instance of the Catalog Class

```
Catalog catalog = new Catalog();
```

```
public void UCAddDVD(String title, String category, Double runtime, int year,
Double price) {
    catalog.addDVD(title, category, runtime, year, price);
}
```

```
require_relative 'catalog'
```

```
class Controller
  def initialize
    @catalog = Catalog.new
  end
```

```
def uc add dvd(title, category, run_time, year, price)
  @catalog.add_dvd(title, category, run_time, year, price)
end
```

pseudocode to ruby

## Circle Application UML to Python

### TUTORIAL 2 on UML to C#, Ruby, Python, PHP5

Build Content

Assessments

Tools

Partner Content



PYTHON and UML to PYTHON

PYTHON



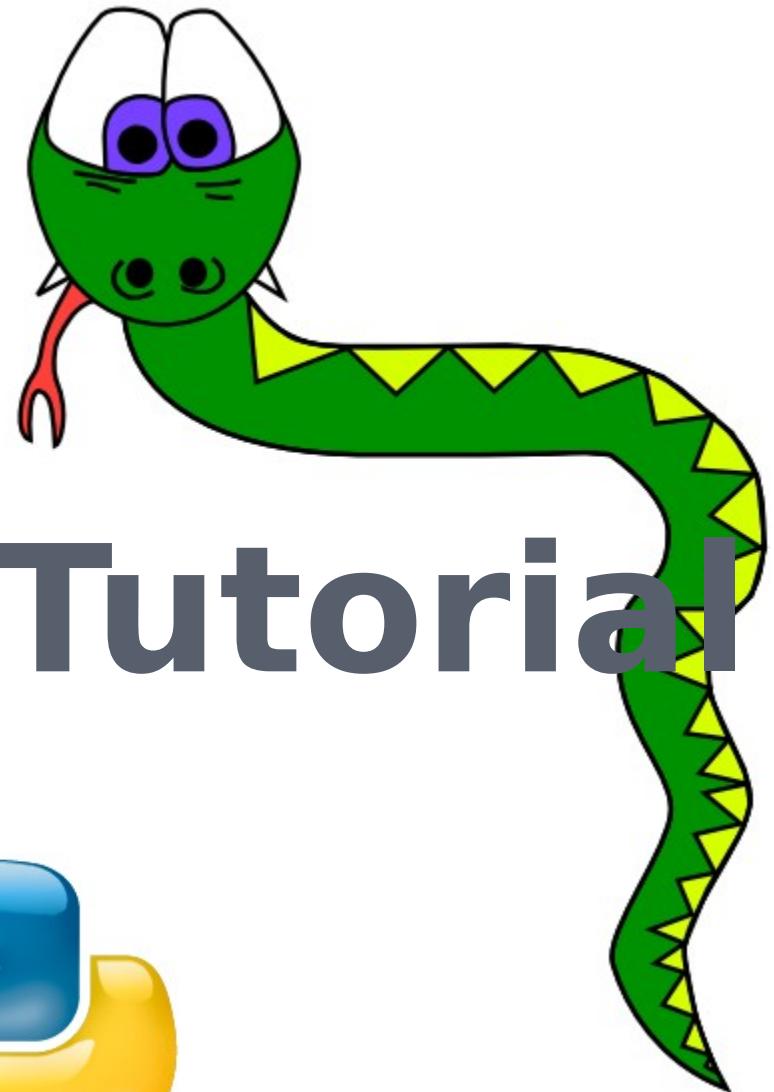
Python Tutorial.pptx

Enabled: Statistics Tracking

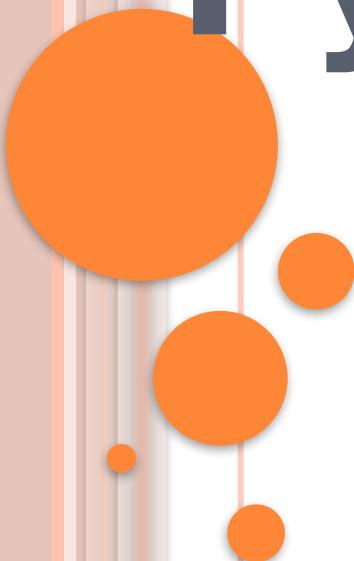
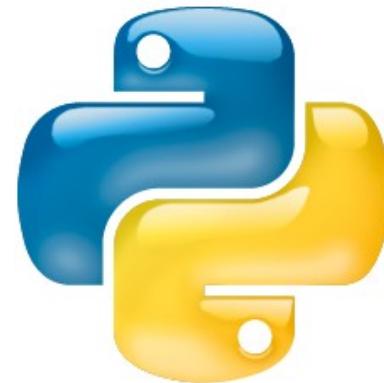


Circle Application UML to Python Step by Step Tutorial.docx

Enabled: Statistics Tracking



# Python Tutorial



## WHAT IS **Python**?

- an **interpreted**, general-purpose high-level programming language whose design philosophy emphasizes code readability
- **O**bject Oriented Programming (**OOP**)
- Python **was conceived in the late 1980s** and its implementation was started in December **1989** by Guido van Rossum at CWI in the Netherlands
- **JPython** is **Java implementation** of Python language

# Hello

# World!

JAVA

```
public class Main
{
    public static void main(String[] args)
    {
        System.out.println("Hello World!");
    }
}
```

C++

```
#include <iostream>
using namespace std;

int main()
{
    cout << "Hello World!" << endl;
}
```



# Hello World!

Python

```
print "Hello World!"
```



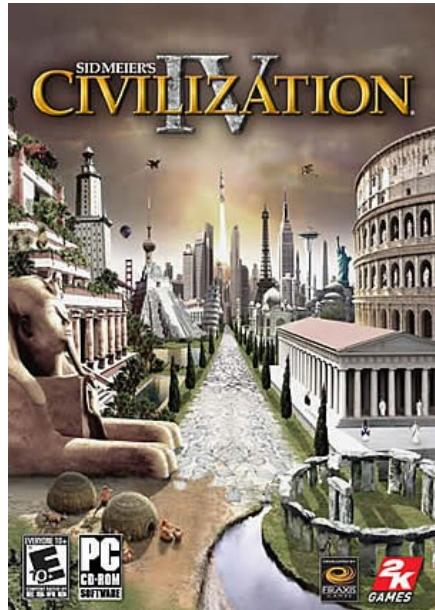
2 lines: Input, assignment

```
name = raw_input('What is your name?\n')
print 'Hi. %s.' % (name)
```

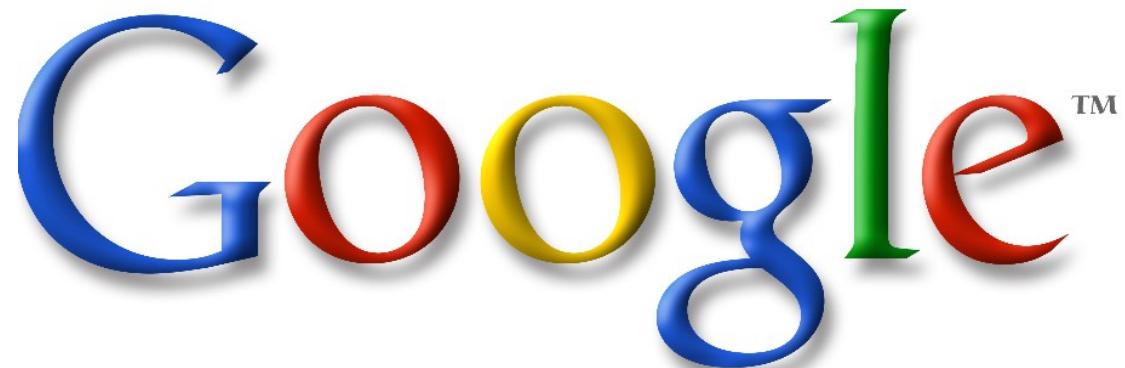
»

..

# WHO USES Python?



Broadcast Yourself™



# Circle Application UML to Python

python - NetBeans IDE 6.9.1

Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Files Services

Python

Sources <Top Level> Point.py python.py

python.py Point.py Circle.py Point.py

```
1 class Point:
2     def __init__(self, x, y):
3         self.__x = x
4         self.__y = y
5
6     def setPoint(self, x, y):
7         self.__x = x
8         self.__y = y
9
10    def getX(self):
11        return self.__x
12
13    def getY(self):
14        return self.__y
15
16    def printPoint(self):
17        print(self.__x, self.__y)
```

Search (Ctrl+I)

Point

- x: double //1
- y: double //2
- + Point (x: double,y: double) //1
- + setPoint (x: double,y: double): void //2
- + getX (): double //3
- + getY (): double //4
- + print (): void //5

# Circle Application UML to Python

NetBeans IDE 6.9.1

File Navigate Source Refactor Run Debug Profile Team Tools Window Help



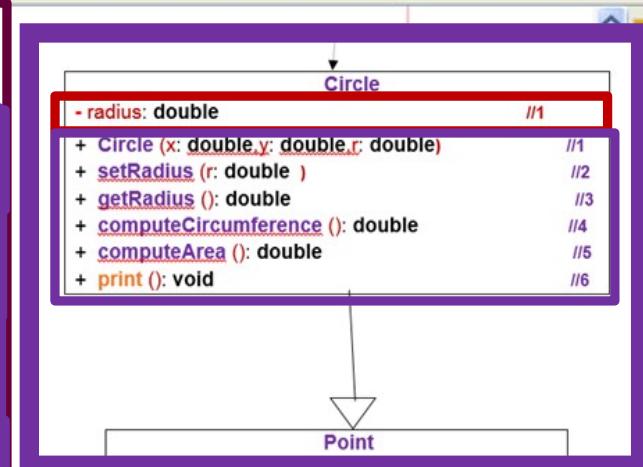
Files Services

Source

<Top Level>  
Circle.py  
Point.py  
python.py

```
1 import Point
2
3 class Circle Point:
4     def __init__(self, x, y, r):
5         Point.__init__(self, x, y)
6         self.__radius = r
7
8     def setRadius(self, r):
9         self.__radius = r
10
11    def getRadius(self):
12        return self.__radius
13
14    def getCircumference(self):
15        return (2 * 3.1416 * self.__radius)
16
17    def getArea(self):
18        return (3.1416 * self.__radius * self.__radius)
19
20    def printCircle(self):
21        print("Center: ")
22        Point.printPoint(self)
23
24
25        print("Radius: ", self.__radius)
26        print("Circumference: ", self.getCircumference())
27        print("Area: ", self.getArea())
28
```

??? inheritance



# Circle Application UML to Python

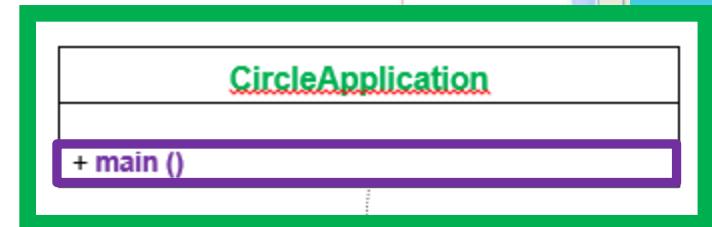
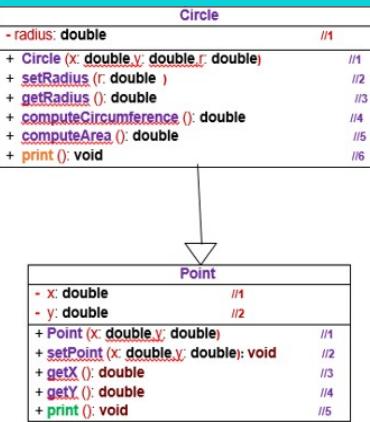
Python - NetBeans IDE 6.9.1

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help



```
1 import Circle
2
3 def main():
4     circle1 = Circle.Circle(3, 2.5, 4)
5     print("***** Circle 1 *****")
6     circle1.printCircle()
7
8
9     circle1.setPoint(-2.5, 7)
10    circle1.setRadius(4)
11    print("***** Circle 1 modified *****")
12    circle1.printCircle()
13
14
15    circle2 = Circle.Circle(0, 0, 0)
16    x = input('Enter x Coordinates of the center:')
17    y = input('Enter y Coordinates of the center:')
18    r = input('Enter radius:')
19    circle2.setPoint(x, y)
20    circle2.setRadius(r)
21    print("***** Circle 2 *****")
22    circle2.printCircle()
23
24 main()
25
```

???



Output - MRT6PythonTutorial

```
***** Circle 1 *****
Center:
(3, 2.5)
('Radius: ', 4)
('Circumference: ', 25.1328)
('Area: ', 50.2656)
***** Circle 2 *****
Center:
(-2.5, 7)
('Radius: ', 4)
('Circumference: ', 25.1328)
('Area: ', 50.2656)
Enter x Coordinates of the center:2
Enter y Coordinates of the center:2
Enter radius:2
***** Circle 3 *****
Center:
(2, 2)
('Radius: ', 2)
('Circumference: ', 12.5664)
('Area: ', 12.5664)
```

# The DVD Collection System

## UML MVC to python



- > \_pycache\_
- Catalog.py
- Catalog.txt
- Controller.py
- DVD.py
- VideoStoreApp.py
- View.py

```

class DVD:
    # Initializer
    def __init__(self, title, category, runTime, year, price):
        self.title = title
        self.category = category
        self.runTime = runTime
        self.year = year
        self.price = price

    def __repr__(self):
        return f'{self.title} {self.category} {self.runTime} \
{self.year} {self.price}'
  
```

pseudocode to python

# The DVD Collection System

## UML MVC to ruby



```

from DVD import DVD
class Catalog:
    def __init__(self):
        self.MAX_DVD = 100
        self.numDVD = 0
        self.dvd = []

    def getIndex(self, title):
        index = None
        for i in range(self.numDVD):
            if(title == self.dvd[i].title):
                index = i
                break
        return index

    def addDVD(self, title, category, runTime, year, price):
        exists = self.getIndex(title)
        if(self.numDVD < self.MAX_DVD and exists == None):
            a = DVD(title, category, runTime, year, price)
            self.dvd.append(a)
            self.numDVD += 1

public void addDVD(String title, String category, Double runtime, int year, Double price) {
    If the title is not already in the catalog
    If(numDVD < MAX_DVD)
        DVD[numDVD] = new DVD(title, category, runtime, year, price);
        numDVD++;
}

```

pseudocode to python

# The DVD Collection System

## UML MVC to ruby



Create the instance of the Catalog Class

```
Catalog catalog = new Catalog();
```

```
public void UCAddDVD(String title, String, category, Double runtime, int year,
Double price) {
    catalog.addDVD(title, category, runtime, year, price);
}
```

```
from Catalog import Catalog
class Controller:
```

```
    def __init__(self):
        self.catalog = Catalog()
```

```
    def UCAddDVD(self, title, category, runTime, year, price):
        self.catalog.addDVD(title, category, runTime, year, price)
```

pseudocode to python

## Circle Application UML to PHP

**TUTORIAL 2 on UML to C#, Ruby, Python, PHP5**

Build Content Assessments Tools Partner Content

 **PHP and UML to PHP**

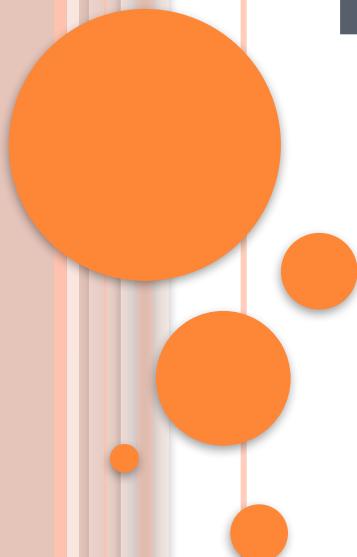
 **PHP NetBeans 6.9.1 mySQL WAMP Step by Step Tutorial.docx**  
Enabled: Statistics Tracking

 **PHP Tutorial.pptx**  
Enabled: Statistics Tracking

 **Circle Application UML to PhP Step by Step Tutorial.docx**  
Enabled: Statistics Tracking

**php5**

# **PHP - Tutorial**



## WHAT IS **PHP**?

- an **interpreted**, general-purpose high-level programming language whose design philosophy emphasizes code readability
- **O**bject **O**riented **P**aradigm (**OOP**)



# Hello

# World!

JAVA

```
public class Main
{
    public static void main(String[] args)
    {
        System.out.println("Hello World!");
    }
}
```

C++

```
#include <iostream>
using namespace std;

int main()
{
    cout << "Hello World!" << endl;
}
```



# Hello World!

**PHP**

```
echo "Hello World!";
```



# Using **objects** in PHP Scripts

- Declare an **object** in PHP by using the **new** operator with a **Class Constructor**
- A **Class Constructor** is a special function with the same name as its **Class** that is called automatically when an **object** from the **Class** is instantiated
- The syntax for instantiating an **object** is:

```
$ObjectName = new ClassName();
```

# Using objects in PHP Scripts

The identifiers for an **object** name:

- ❑ Must begin with a dollar sign
- ❑ Can include numbers or an underscore
- ❑ Cannot include spaces
- ❑ Are case sensitive

```
$Checking = new BankAccount();
```

- ❑ Can pass arguments to many **Constructor functions**

```
$Checking = new BankAccount(01234587, 1021, 97.58);
```

# Using objects in PHP Scripts

- With **methods**, include a set of parentheses at the end of the **method** name, just as with **functions**
- Like functions, **methods** can also accept arguments

```
$Checking->getBalance();  
$CheckNumber = 1022;  
$Checking->getCheckAmount($CheckNumber);
```

# Circle Application UML to PHP

PhP - NetBeans IDE 6.9.1

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

index.php x Point.php x Circle.php x

Search (Ctrl+I)

Files Services

Source Files

Circle.php Point.php index.php

Include Path

```
<?php
class Point {
    private $x, $y;

    function __construct($x, $y) {
        $this->x = $x;
        $this->y = $y;
    }

    function setPoint($x, $y) {
        $this->x = $x;
        $this->y = $y;
    }

    function getX() {
        return $this->x;
    }

    function getY() {
        return $this->y;
    }

    function printPoint() {
        echo '[' . $this->x . ', ' . $this->y . ']'. '<br />';
    }
}
```

Point

- x: double //1
- y: double //2
- + Point (x: double,y: double) //1
- + setPoint (x: double,y: double): void //2
- + getX (): double //3
- + getY (): double //4
- + print (): void //5

# Circle Application UML to PHP

PHP - NetBeans IDE 6.9.1

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

index.php x Point.php x Circle.php x

Search (Ctrl+I)

Files Services

Source Files

Circle.php Point.php index.php

Include Path

```
<?php
    include ("Point");
    class Circle extends Point {
        private $x, $y, $radius;
        function __construct($x, $y, $r) {
            $this->radius = $r;
            parent::__construct($x, $y);
        }
        function setRadius($r){
            $this->radius = $r;
        }
        function getRadius(){
            return $this->radius;
        }
        function getCircumference(){
            return (2 * 3.1416 * $this->radius);
        }
        function getArea(){
            return (3.1416 * $this->radius * $this->radius);
        }
        function printCircle(){
            echo 'Center: ';
            $this->printPoint();
        }
    }

```

??? inheritance

Circle

+ radius: double //1

+ Circle (x: double,y: double,r: double) //1

+ setRadius (r: double ) //2

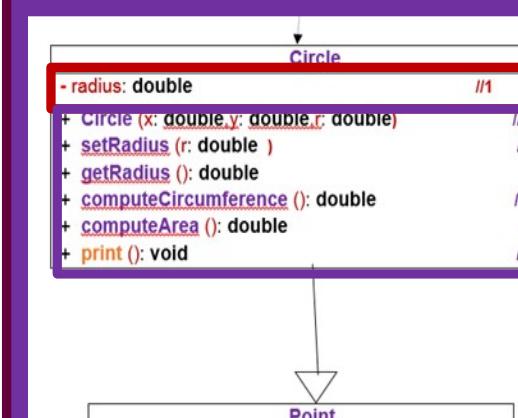
+ getRadius () : double //3

+ computeCircumference () : double //4

+ computeArea () : double //5

+ print () : void //6

Point



```
graph TD
    Circle --> Point
```

# Circle Application UML to PhP

PhP - NetBeans IDE 6.9.1

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Files Services

index.php Point.php Circle.php

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<title></title>
<?php
include ("Circle.php");
?>
</head>
<body>
<?php
$circle1 = new Circle(3, 2.5, 4);
echo '***** Circle 1 *****' . '<br />';
$circle1->printCircle();

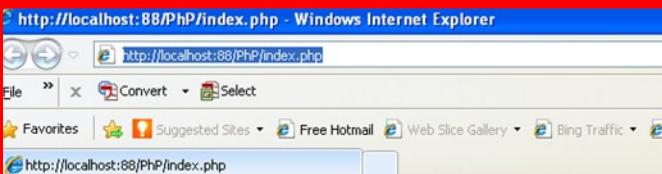
$circle1->setPoint(-2.5, 7);
$circle1->setRadius(4);
echo '<br />' . '***** Circle 1 Modified *****' . '<br />';
$circle1->printCircle();

echo '<br />' . '***** Circle 3 *****' . '<br />';
$circle2 = new Circle(0, 0, 0);

echo '<br />' . 'Enter x Coordinates of the center: ';
$x = 2;
echo '<br />' . 'Enter y Coordinates of the center: ';
$y = 2;
echo '<br />' . 'Enter radius: ';
$r = 2;
$circle2->setPoint($x, $y);
$circle2->setRadius($r);
$circle2->printCircle();
```

```
Circle
- radius: double //1
+ Circle(x: double, y: double, r: double) //2
+ setRadius(r: double) //3
+ getRadius(): double //4
+ computeCircumference(): double //5
+ computeArea(): double //6
+ print(): void //6

Point
- x: double //1
- y: double //2
+ Point(x: double, y: double) //1
+ setPoint(x: double, y: double): void //2
+ getX(): double //3
+ getY(): double //4
+ print(): void //5
```



\*\*\*\*\* Circle 1 \*\*\*\*\*  
Center: [3 , 2.5]  
Radius: 4  
Circumference: 25.1328  
Area: 50.2656

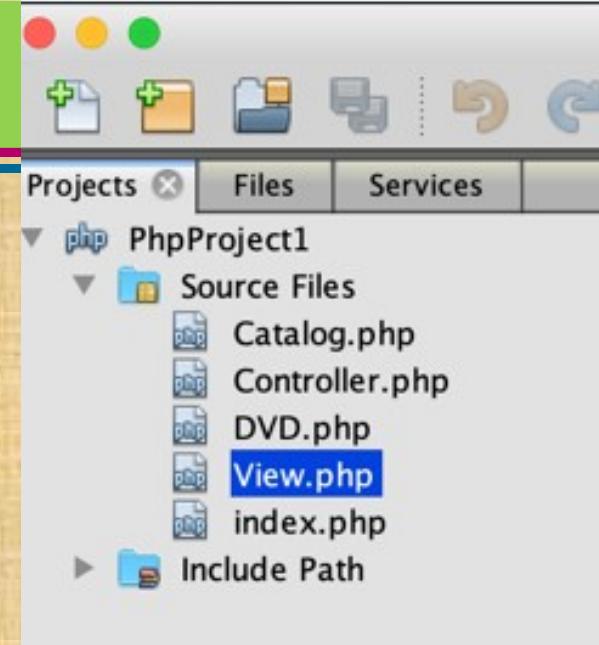
\*\*\*\*\* Circle 1 Modified \*\*\*\*\*  
Center: [-2.5 , 7]  
Radius: 4  
Circumference: 25.1328  
Area: 50.2656

\*\*\*\*\* Circle 3 \*\*\*\*\*

Enter x Coordinates of the center:  
Enter y Coordinates of the center:  
Enter radius:Center: [2 , 2]  
Radius: 2  
Circumference: 12.5664  
Area: 12.5664

# The DVD Collection System

## UML MVC to php



```

<?php

class DVD {
    private $title, $category, $runtime, $year, $price;

    function __construct($title, $category, $runtime, $year, $price)
    {
        [$this->title = $title, $this->category = $category, $this->runtime =
$runtime,
         $this->year = $year, $this->price = $price];
    }
}
  
```

pseudocode to php

# The DVD Collection System

## UML MVC to php



```

<?php
require 'DVD.php';

class Catalog {
    const MAX_DVD = 100;
    private $numDVD, $dvd;

    function __construct() {
        $this->numDVD = 0;
        $this->dvd = [];
    }

    function addDVD($title, $category, $runtime, $year, $price) {
        $exists = 0;
        for ($index = 0; $index < $this->numDVD; $index++) {
            if ($title == $this->dvd[$index][0]) {
                $exists = 1;
            }
        }
        if ($this->numDVD < MAX_DVD && $exists != 1) {
            $this->dvd[$this->numDVD] = new DVD($title, $category, $runtime,
                $year, $price);
            $this->numDVD++;
        }
    }

    public void addDVD(String title, String category, Double runtime, int year, Double
        price) {
        If the title is not already in the catalog
        If(numDVD < MAX_DVD)
            DVD[numDVD] = new DVD(title, category, runtime, year, price);
            numDVD++;
    }
}
  
```

pseudocode to php

# The DVD Collection System

## UML MVC to php



Create the instance of the Catalog Class

```
Catalog catalog = new Catalog();
```

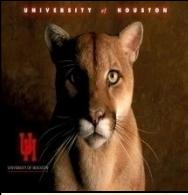
```
public void UCAddDVD(String title, String category, Double runtime, int year,
Double price) {
    catalog.addDVD(title, category, runtime, year, price);
}
```

```
<?php
require 'Catalog.php';
```

```
class Controller {
    private $catalog;
    function __construct() {
        $this->catalog = new Catalog();
    }
}
```

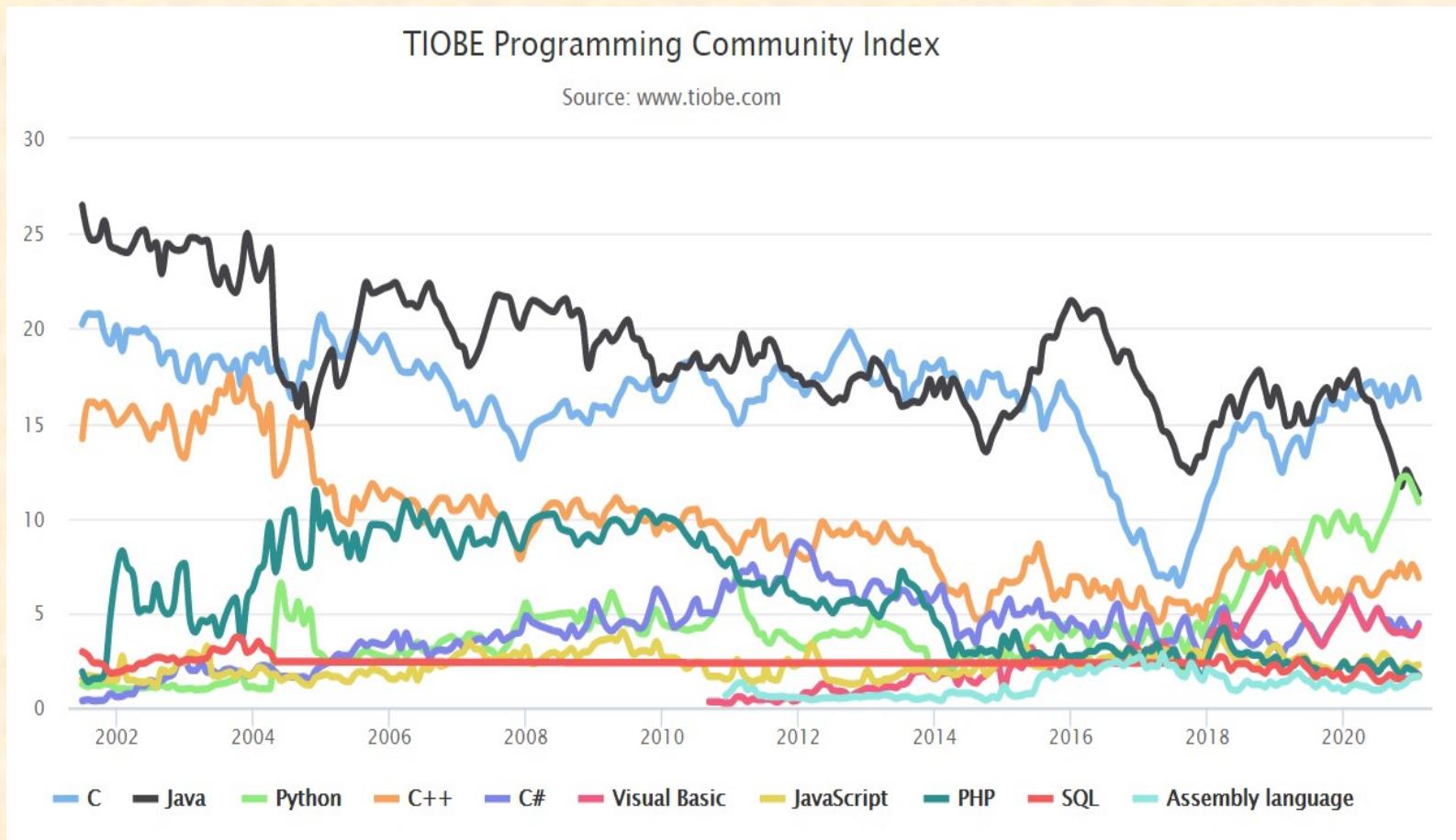
```
function UCAddDVD($title, $category, $runtime, $year, $price) {
    $this->catalog->addDVD($title, $category, $runtime, $year, $price);
}
```

pseudocode to php



# I love UML

C#, RUBY, PYTHON, PHP  
One to one mapping



09.25.2023 (M 4 to 5:30) (10)	<p>Web App Dev:</p> <ul style="list-style-type: none"><li>1. Guide to Web Application Development 2015.pdf</li><li>2. PHP vs ASP.net Comparison 2015.pdf</li><li>3. Ruby on Rails vs PHP Comparison 2015.pdf</li></ul>	<p>Tutorials 2 on Front End Languages C#, RUBY, PYTHON, PHP</p>	<p>OO Languages CANVAS Assignment</p>	
09.27.2023 (W 4 to 5:30) (11)	<p>Estimating and Planning:</p> <ul style="list-style-type: none"><li>1. Estimating Techniques 2015.pdf</li><li>2. How do you estimate on an Agile project 2015.pdf</li></ul>	<p>Tutorials 3 on C# Visual Studio IDE MVC, PYTHON PyCharm IDE Django, PHP PhpStorm IDE Zend frameworks WE CANVAS APPS</p>	<p>WEB APP Papers Summary (1 Page) CANVAS Assignment</p>	

09.25.2023 (M 4 to 5:30)  (10)	<p><b>Web App Dev:</b></p> <ol style="list-style-type: none"><li>1. Guide to Web Application Development 2015.pdf</li><li>2. PHP vs ASP.net Comparison 2015.pdf</li><li>3. Ruby on Rails vs PHP Comparison 2015.pdf</li></ol>	<p>Tutorials 2 on Front End Languages C#, RUBY, PYTHON, PHP</p>	<p>OO Languages CANVAS Assignment</p>
---	---	---	---

CLASSES

PAPERS on WEB APP DEV	VH– Publish.
Guide to Web Application Development.pdf	VH– Publish.
PHP vs ASP.net Comparison.pdf	VH– Publish.
Ruby on Rails vs PHP Comparison.pdf	VH– Publish.

One Page Summary of Web App Dev Papers

• 12 pt. font, double spaced, MLA format, no citations required, times new roman.  
• Introduction, body, and conclusion paragraphs.  
• Submit as PDF or DOCX file.

Please attach your electronic version.

**For the Grader:** if the student did not submit, please skip and do not assign a ZERO (MISSING).

VH– Publish.

**From 5:05 to 5:15 PM – 10 minutes.**

09.25.2023 (M 4 to 5:30)  (10)	<b>Web App Dev:</b> 1. Guide to Web Application Development 2015.pdf 2. PHP vs ASP.net Comparison 2015.pdf 3. Ruby on Rails vs PHP Comparison 2015.pdf	<b>Tutorials 2 on Front End Languages C#, RUBY, PYTHON, PHP</b>	<b>OO Languages</b> <b>CANVAS Assignment</b>	
---	---	---	---	--

CLASS PARTICIPATION 20 points

20% of Total + :

## PASSWORD: IN TEAMS

END Class 10 Participation

CLASS PARTICIPATION 20% Module | Not available until Sep 25 at 5:05pm | Due Sep 25 at 5:15pm | 100 pts

**At 5:15 PM.**

**End Class 10**

**VH, Download Attendance Report  
Rename it:  
9.25.2023 Attendance Report FINAL**

**VH, upload Class 10 to CANVAS**