

<https://www.tutorialsteacher.com/csharp/csharp-data-types>

<https://www.c-sharpcorner.com/article/datetime-in-c-sharp/>

Constructors in Inheritance

5

Public Def

Pri

Public Para

Public Copy

Public static

Class A{

Id , name;

**Public A() {}**

**A() {}**

**Public A(int id, string name){}**

**Public void Get() {}**

**Public void Display() {}**

**}**

Class B : A

{

Get()

Display()

**// CONSTUCTORS ARE NEVER INHERITED TO THE CHILD CLASS**

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace A1

{

class Employee

{

int id;

string name;

string manager;

public Employee()

{

}

public Employee(int id, string name, string manager

)

{

this.id = id;

this.name = name;

this.manager = manager;

}

public void GetDetails()

{

Console.WriteLine("Enter ID");

id = int.Parse(Console.ReadLine());

Console.WriteLine("Enter Name");

name = Console.ReadLine();

Console.WriteLine("Enter Manager Name");

manager = Console.ReadLine();

}

public void DisplayDetails()

{

Console.WriteLine("ID is " + id);

Console.WriteLine("Name is " + name);

Console.WriteLine("Manager Name is " + manager);

}

}

class PartTimeEmployee : Employee

{

int commercials;

string Duration;

**public PartTimeEmployee() :base ()**

**{**

**}**

**public PartTimeEmployee(int id, string name, string manager,**

**int commercials, string duration) :base (id,name,manager)**

**{**

**this.commercials = commercials;**

**this.Duration = duration;**

**}**

public void GetDetails()

{

base.GetDetails();

Console.WriteLine("Enter commercials");

commercials = int.Parse(Console.ReadLine());

Console.WriteLine("Enter Duration");

Duration = Console.ReadLine();

}

public void DisplayDetails()

{

base.DisplayDetails();

Console.WriteLine("commercials is " + commercials);

Console.WriteLine("Duration is " + Duration);

}

}

class FullTimeEmployee : Employee

{

int salary;

public FullTimeEmployee()

{

}

**public FullTimeEmployee(int id, string name, string manager,**

**int salary) : base(id,name,manager)**

**{**

**this.salary = salary;**

**}**

public void GetDetails()

{

base.GetDetails();

Console.WriteLine("Enter salary");

salary = int.Parse(Console.ReadLine());

}

public void DisplayDetails()

{

base.DisplayDetails();

Console.WriteLine("Salary is " + salary);

}

}

class Program2

{

static void Main()

{

//Console.WriteLine("Employee Objet");

//Employee employee = new Employee();

//employee.GetDetails();

//employee.DisplayDetails();

Console.WriteLine("Parttime Employee Object");

PartTimeEmployee partTimeEmployee = new PartTimeEmployee();

partTimeEmployee.GetDetails();

partTimeEmployee.DisplayDetails();

PartTimeEmployee partTimeEmployee1 = new PartTimeEmployee(

100, "AJay", "Deepak", 12000, "12 Months");

FullTimeEmployee fullTimeEmployee = new FullTimeEmployee();

fullTimeEmployee.GetDetails();

fullTimeEmployee.DisplayDetails();

FullTimeEmployee fullTimeEmployee1 = new FullTimeEmployee(

102, "Karan", "Mandeep", 12000);

}

}

}

When we create object of child class it always invokes constructor of the parent class first

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Base Class Members | Private | Public | Protected |  |
|  | NO  NEVER ARE INHERITED | YES , AND THEY WILL BE ALSO ACCESSIBLE OUTSIDE THE CLASS THROUGH THE OBJECT | THEY ARE INHERITED TO THE CHILD CLASS, BUT ARE NOT ACCESSIBLE OUTSIDE THE CLASS |  |

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ClassDemo

{

class A

{

int x;

public int y;

protected int z;

}

class B : A

{

int b;

void cal()

{

y++;

z++;

}

}

class Demo

{

static void Main()

{

A a = new A();

B b = new B();

b.y = 10;

//b.z = 100; z is not accessible outside the class

}

}

}

Student Class  
Rn

Name

Sports Class

Sportsname

Score

Test Class

Marks in 3 subjects

Result Class

Calculate Result in this class

TOTAL SCORE = MARKS IN 3 SUBJECTS + SCORE OBTAINED IN SPORTS

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace SMSProject

{

class Student

{

int rn;

string name;

string address;

public void GetDetails()

{

Console.WriteLine("Enter Roll No");

rn = Convert.ToByte(Console.ReadLine());

Console.WriteLine("Enter Name");

name = Console.ReadLine();

Console.WriteLine("Enter Address");

address = Console.ReadLine();

}

public void DisplayDetails()

{

Console.WriteLine("Roll No is " + rn);

Console.WriteLine("Name is " + name);

Console.WriteLine("Address is " + address);

}

}

class Sports : Student

{

string sportsName;

protected int score;

public void GetDetails()

{

base.GetDetails();

Console.WriteLine("Enter Sports Name");

sportsName = Console.ReadLine();

Console.WriteLine("Enter Score");

score = int.Parse(Console.ReadLine());

}

public void DisplayDetails()

{

base.DisplayDetails();

Console.WriteLine("Sports Name is " + sportsName );

Console.WriteLine("Score is " + score);

}

}

class Test : Sports

{

int m1, m2, m3;

protected int total\_marks;

public void GetDetails()

{

base.GetDetails();

Console.WriteLine("Enter Marks1");

m1 = Byte.Parse(Console.ReadLine());

Console.WriteLine("Enter Marks2");

m2 = Byte.Parse(Console.ReadLine());

Console.WriteLine("Enter Marks3");

m3 = Byte.Parse(Console.ReadLine());

total\_marks = m1 + m2 + m3;

}

public void DisplatDetails()

{

base.DisplayDetails();

Console.WriteLine("Total Marks are " + total\_marks);

}

}

class Result : Test

{ int total\_score ;

public void CalculateTotalMarks()

{

total\_score = total\_marks + score;

}

}

class A

{

static void Main()

{

Result res = new Result();

res.GetDetails();

res.CalculateTotalMarks();

res.DisplayDetails();

}

}

}