

DEAKIN UNIVERSITY

OBJECT ORIENTED DEVELOPMENT

ONTRACK SUBMISSION

C# Essentials: Inheritance

Submitted By:

Peter STACEY

pstacey

2020/04/06 13:00

Tutor:

Dipto PRATYAKSA

Outcome	Weight
Evaluate Code	◆◆◆◆◆
Principles	◆◆◆◆◆
Build Programs	◆◆◆◆◆
Design	◆◆◆◆◆
Justify	◆◆◆◆◆

This task introduced inheritance, one of the key principles that help define object oriented programming. The task also begins to introduce the concept of polymorphism, although it primarily sticks with inheritance as the major theme. The task involves implementing a set design and evaluating our implementation for correct use of conventions.

April 6, 2020



```
1  using System;
2
3  namespace Task01
4  {
5      class ZooPark
6      {
7          static void Main(string[] args)
8          {
9              Animal williamWolf = new Animal("William the Wolf", "Meat", "Dog
10                 ↳ Village", 50.6, 9, "Grey");
11              Animal tonyTiger = new Animal("Tony the Tiger", "Meat", "Cat Land",
12                 ↳ 110, 6, "Orange and White");
13              Animal edgarEagle = new Animal("Edgar the Eagle", "Fish", "Bird Mania",
14                 ↳ 20, 15, "Black");
15          }
16      }
17  }
```

```
1  using System;
2
3  namespace Task01
4  {
5      class Animal
6      {
7          // Instance variables
8          private String _name;
9          private String _diet;
10         private String _location;
11         private double _weight;
12         private int _age;
13         private String _colour;
14
15         /// <summary>
16         /// Constructor for an animal
17         /// </summary>
18         /// <param name="name">The personal name of the animal</param>
19         /// <param name="diet">The primary type of food eaten</param>
20         /// <param name="location">The exhibition the animal is in</param>
21         /// <param name="weight">Weight in pounds</param>
22         /// <param name="age">Age of the animal in years</param>
23         /// <param name="colour">The dominant color(s)</param>
24         public Animal(String name, String diet, String location,
25             double weight, int age, String colour)
26         {
27             _name = name;
28             _diet = diet;
29             _location = location;
30             _weight = weight;
31             _age = age;
32             _colour = colour;
33         }
34
35         /// <summary>
36         /// Method to make the animal eat food
37         /// </summary>
38         public void eat()
39         {
40             // Code for the animal to eat
41             Console.WriteLine("The animal eats food");
42         }
43
44         /// <summary>
45         /// Method to make the animal sleep
46         /// </summary>
47         public void sleep()
48         {
49             // Code for the animal to sleep
50             Console.WriteLine("The animal sleeps");
51         }
52
53         /// <summary>
```

```
54     /// Method to make the animal make a noise
55     /// </summary>
56     public void makeNoise()
57     {
58         // Code for the animal to make a noise
59         Console.WriteLine("The animal makes a noise");
60     }
61
62     /// <summary>
63     /// Method to make any animal sound like a lion
64     /// </summary>
65     public void makeLionNoise()
66     {
67         // Code for the animal to make a noise
68         Console.WriteLine("The Lion makes a noise");
69     }
70
71     /// <summary>
72     /// Method to make any animal sound like an eagle
73     /// </summary>
74     public void makeEagleNoise()
75     {
76         // Code for the animal to make a noise
77         Console.WriteLine("The eagle makes a noise");
78     }
79
80     /// <summary>
81     /// Method to make any animal sound like a wolf
82     /// </summary>
83     public void makeWolfNoise()
84     {
85         // Code for the animal to make a noise
86         Console.WriteLine("The wold makes a noise");
87     }
88
89     /// <summary>
90     /// Method to make an animal eat meat
91     /// </summary>
92     public void eatMeat()
93     {
94         // Code for the animal to make a noise
95         Console.WriteLine("The animal eats meat");
96     }
97
98     /// <summary>
99     /// Method to make an animal eat berries
100    /// </summary>
101    public void eatBerries()
102    {
103        // Code for the animal to make a noise
104        Console.WriteLine("The animal eats berries");
105    }
106
```

```
107      /// <summary>
108      /// Method to make an animal eat fish
109      /// </summary>
110      public void eatFish()
111      {
112          // Code for the animal to make a noise
113          Console.WriteLine("The animal eats fish");
114      }
115  }
116 }
```

```
1  using System;
2
3  namespace Task02
4  {
5      /// <summary>
6      /// Base class for all animals
7      /// </summary>
8      class Animal
9      {
10         // Instance variables
11         private String _name;
12         private String _diet;
13         private String _location;
14         private double _weight;
15         private int _age;
16         private String _colour;
17
18         // Public properties
19         public String Name { get => _name; }
20
21         /// <summary>
22         /// Constructor for a base animal instance
23         /// </summary>
24         /// <param name="name">The personal name of the animal</param>
25         /// <param name="diet">The primary type of food eaten</param>
26         /// <param name="location">The exhibition the animal is in</param>
27         /// <param name="weight">Weight in pounds</param>
28         /// <param name="age">Age of the animal in years</param>
29         /// <param name="colour">The dominant color(s)</param>
30         public Animal(String name, String diet, String location,
31             double weight, int age, String colour)
32         {
33             _name = name;
34             _diet = diet;
35             _location = location;
36             _weight = weight;
37             _age = age;
38             _colour = colour;
39         }
40
41         /// <summary>
42         /// Method to make the animal eat food
43         /// </summary>
44         public virtual void eat()
45         {
46             // code for animal to eat
47             Console.WriteLine("An animal eats");
48         }
49
50         /// <summary>
51         /// Puts the animal to sleep
52         /// </summary>
53         public virtual void sleep()
```

```
54     {
55         // code for animal to sleep
56         Console.WriteLine("An animal sleeps");
57     }
58
59     /// <summary>
60     /// Allows the animal to speak or make noise
61     /// </summary>
62     public virtual void makeNoise()
63     {
64         // code for animal to make a noise
65         Console.WriteLine("An animal makes a noise");
66     }
67
68     /// <summary>
69     /// Allows the animal to construct it's home within the display
70     /// </summary>
71     public virtual void buildHome()
72     {
73         Console.WriteLine("An animal builds a home");
74     }
75 }
76 }
```

```
1  using System;
2
3  namespace Task02
4  {
5      class ZooPark
6      {
7          static void Main(string[] args)
8          {
9              //Animal williamWolf = new Animal("William the Wolf", "Meat", "Dog
              ↳ Village", 50.6, 9, "Grey");
10             //Animal tonyTiger = new Animal("Tony the Tiger", "Meat", "Cat Land",
              ↳ 110, 6, "Orange and White");
11             //Animal edgarEagle = new Animal("Edgar the Eagle", "Fish", "Bird
              ↳ Mania", 20, 15, "Black");
12
13             Tiger tonyTiger = new Tiger("Tony the Tiger", "Meat", "Cat Land", 110,
              ↳ 6,
14                 "Orange and White", "Siberian", "White");
15             Wolf williamWolf = new Wolf("William the Wolf", "Meat", "Dog Village",
              ↳ 50.6, 9, "Grey");
16             Eagle edgarEagle = new Eagle("Edgar the Eagle", "Fish", "Bird Mania",
              ↳ 20, 15,
17                 "Black", "Harpy", 98.5);
18
19             Animal baseAnimal = new Animal("Animal Name", "Animal Diet", "Animal
              ↳ Location",
20                 0.0, 0, "Animal Colour");
21
22             baseAnimal.eat();
23             tonyTiger.eat();
24             williamWolf.eat();
25             edgarEagle.eat();
26
27             baseAnimal.sleep();
28             tonyTiger.sleep();
29             williamWolf.sleep();
30             edgarEagle.sleep();
31
32             baseAnimal.makeNoise();
33             tonyTiger.makeNoise();
34             williamWolf.makeNoise();
35             edgarEagle.makeNoise();
36
37             baseAnimal.buildHome();
38             tonyTiger.buildHome();
39             williamWolf.buildHome();
40             edgarEagle.buildHome();
41
42             edgarEagle.layEgg();
43             edgarEagle.fly();
44
45             Lion leoLion = new Lion("Leo the Lion", "Meat", "Lion's Pride", 145, 3,
              ↳ "Sandy", "African");
```



```
46         Penguin percyPenguin = new Penguin("Percy the Penguin", "Fish",
47             ↪ "Antarctic Experience",
48             12, 2, "Black and White", "Emperor", 20);
49
49         leoLion.eat();
50         leoLion.makeNoise();
51         leoLion.buildHome();
52         leoLion.sleep();
53
54         percyPenguin.eat();
55         percyPenguin.buildHome();
56         percyPenguin.layEgg();
57         percyPenguin.makeNoise();
58         percyPenguin.fly();
59
60         Wolf walterWolf = new Wolf("Walter the Wolf", "Meat", "Dog Village",
61             ↪ 45.5, 5, "Brown");
62
62         williamWolf.makeNoise();
63         walterWolf.makeNoise();
64         williamWolf.buildHome();
65         walterWolf.sleep();
66     }
67 }
68 }
```

```
1  using System;
2
3  namespace Task02
4  {
5      /// <summary>
6      /// Prototype for a tiger as a type of feline
7      /// </summary>
8      class Tiger : Feline
9      {
10         private String _colourStripes;
11
12         /// <summary>
13         /// Constructor for a tiger as a type of feline
14         /// </summary>
15         /// <param name="name">The personal name of the tiger</param>
16         /// <param name="diet">The primary type of food eaten</param>
17         /// <param name="location">The exhibition the tiger is in</param>
18         /// <param name="weight">Weight in pounds</param>
19         /// <param name="age">Age of the tiger in years</param>
20         /// <param name="colour">The dominant color(s)</param>
21         /// <param name="species">The species of tiger</param>
22         public Tiger(String name, String diet, String location,
23             double weight, int age, String colour, String species,
24             String colourStripes)
25             : base(name, diet, location, weight, age, colour, species)
26         {
27             _colourStripes = colourStripes;
28         }
29
30         /// <summary>
31         /// The tiger eats meat
32         /// </summary>
33         public override void eat()
34         {
35             Console.WriteLine("{0}, eats 20lbs of meat", Name);
36         }
37
38         /// <summary>
39         /// The tiger roars
40         /// </summary>
41         public override void makeNoise()
42         {
43             Console.WriteLine("ROARRRRRRRRRRR");
44         }
45
46         /// <summary>
47         /// The tiger makes it's home in the display
48         /// </summary>
49         public override void buildHome()
50         {
51             Console.WriteLine("{0} builds a lair", Name);
52         }
53     }
```

54 }

```
1  using System;
2
3  namespace Task02
4  {
5      /// <summary>
6      /// Prototype for an eagle as a type of bird
7      /// </summary>
8      class Eagle : Bird
9      {
10
11         /// <summary>
12         /// Constructor for an eagle as a type of bird
13         /// </summary>
14         /// <param name="name">The personal name of the eagle</param>
15         /// <param name="diet">The primary type of food eaten</param>
16         /// <param name="location">The exhibition the eagle is in</param>
17         /// <param name="weight">Weight in pounds</param>
18         /// <param name="age">Age of the eagle in years</param>
19         /// <param name="colour">The dominant color(s)</param>
20         /// <param name="species">The species of bird</param>
21         /// <param name="wingspan">The wingspan in centimetres</param>
22         public Eagle(String name, String diet, String location,
23             double weight, int age, String colour, String species,
24             double wingSpan)
25             : base(name, diet, location, weight, age, colour, species, wingSpan)
26         { }
27
28         /// <summary>
29         /// Allows the eagle to roost in it's nest by laying an egg
30         /// </summary>
31         public void layEgg()
32         {
33             // code to allow eagles to lay eggs
34             Console.WriteLine("{0} lays an egg. That's a feat of evolution", Name);
35         }
36
37         /// <summary>
38         /// The eagle flies
39         /// </summary>
40         public override void fly()
41         {
42             // code to allow eagles to fly
43             Console.WriteLine("{0} spreads his wings and flies", Name);
44         }
45
46         /// <summary>
47         /// The eagle eats food
48         /// </summary>
49         public override void eat()
50         {
51             Console.WriteLine("{0} eats 1lb of fish", Name);
52         }
53     }
```

```
54     /// <summary>
55     /// The eagle sleeps in it's nest
56     /// </summary>
57     public override void sleep()
58     {
59         Console.WriteLine("{0} rests in his nest, asleep", Name);
60     }
61
62     /// <summary>
63     /// The eagle squarks
64     /// </summary>
65     public override void makeNoise()
66     {
67         Console.WriteLine("{0} squarks", Name);
68     }
69
70     /// <summary>
71     /// The eagle builds it's nest
72     /// </summary>
73     public override void buildHome()
74     {
75         Console.WriteLine("{0} builds a nest", Name);
76     }
77 }
78 }
```

```
1  using System;
2
3  namespace Task02
4  {
5      class Wolf : Animal
6      {
7          /// <summary>
8          /// Constructor for a wolf as a type of animal
9          /// </summary>
10         /// <param name="name">The personal name of the wolf</param>
11         /// <param name="diet">The primary type of food eaten</param>
12         /// <param name="location">The exhibition the wolf is in</param>
13         /// <param name="weight">Weight in pounds</param>
14         /// <param name="age">Age of the wolf in years</param>
15         /// <param name="colour">The dominant color(s)</param>
16         public Wolf(String name, String diet, String location,
17             double weight, int age, String colour)
18             : base(name, diet, location, weight, age, colour)
19         { }
20
21         /// <summary>
22         /// The wolf eats meat
23         /// </summary>
24         public override void eat()
25         {
26             Console.WriteLine("{0} eats 10lbs of meat", Name);
27         }
28
29         /// <summary>
30         /// The wolf sleeps
31         /// </summary>
32         public override void sleep()
33         {
34             Console.WriteLine("{0} settles down in his den and sleeps", Name);
35         }
36
37         /// <summary>
38         /// The wolf howls
39         /// </summary>
40         public override void makeNoise()
41         {
42             Console.WriteLine("{0} howls", Name);
43         }
44
45         /// <summary>
46         /// The wolf makes it's den
47         /// </summary>
48         public override void buildHome()
49         {
50             Console.WriteLine("{0} builds a den", Name);
51         }
52     }
53 }
```

```
1  using System;
2
3  namespace Task02
4  {
5      /// <summary>
6      /// Super class for all cats as a type of animal
7      class Feline : Animal
8      {
9          private String _species;
10
11         /// <summary>
12         /// Constructor for an eagle as a type of feline
13         /// </summary>
14         /// <param name="name">The personal name of the feline</param>
15         /// <param name="diet">The primary type of food eaten</param>
16         /// <param name="location">The exhibition the feline is in</param>
17         /// <param name="weight">Weight in pounds</param>
18         /// <param name="age">Age of the feline in years</param>
19         /// <param name="colour">The dominant color(s)</param>
20         /// <param name="species">The species of feline</param>
21         public Feline(String name, String diet, String location,
22             double weight, int age, String colour, String species)
23             : base(name, diet, location, weight, age, colour)
24         {
25             _species = species;
26         }
27
28         /// <summary>
29         /// Allows a cat to sleep
30         /// </summary>
31         public override void sleep()
32         {
33             Console.WriteLine("{0} lays down and goes to sleep", Name);
34         }
35     }
36 }
```

```
1  using System;
2
3  namespace Task02
4  {
5      /// <summary>
6      /// Prototype for a lion as a type of feline
7      /// </summary>
8      class Lion : Feline
9      {
10         /// <summary>
11         /// Constructor for a lion as a type of feline
12         /// </summary>
13         /// <param name="name">The personal name of the lion</param>
14         /// <param name="diet">The primary type of food eaten</param>
15         /// <param name="location">The exhibition the lion is in</param>
16         /// <param name="weight">Weight in pounds</param>
17         /// <param name="age">Age of the lion in years</param>
18         /// <param name="colour">The dominant color(s)</param>
19         /// <param name="species">The species of lion</param>
20         public Lion(String name, String diet, String location,
21             double weight, int age, String colour, String species)
22             : base(name, diet, location, weight, age, colour, species)
23         { }
24
25         /// <summary>
26         /// The lion eats
27         /// </summary>
28         public override void eat()
29         {
30             Console.WriteLine("{0} eats 50lbs of meat", Name);
31         }
32
33         /// <summary>
34         /// The lion roars bigly
35         /// </summary>
36         public override void makeNoise()
37         {
38             Console.WriteLine("BIIIIIGGGG ROARRRRRRRRRRRR");
39         }
40
41         /// <summary>
42         /// The lion builds the location for it's pride in the display
43         /// </summary>
44         public override void buildHome()
45         {
46             Console.WriteLine("{0} builds a den", Name);
47         }
48     }
49 }
```



```
1  using System;
2
3  namespace Task02
4  {
5      /// <summary>
6      /// Prototype for a bird as type of animal
7      /// </summary>
8      class Bird : Animal
9      {
10         // Instance variables
11         private String _species;
12         private double _wingSpan;
13
14         /// <summary>
15         /// Constructor for a bird instance
16         /// </summary>
17         /// <param name="name">The personal name of the bird</param>
18         /// <param name="diet">The primary type of food eaten</param>
19         /// <param name="location">The exhibition the bird is in</param>
20         /// <param name="weight">Weight in pounds</param>
21         /// <param name="age">Age of the bird in years</param>
22         /// <param name="colour">The dominant color(s)</param>
23         /// <param name="species">The species of bird</param>
24         /// <param name="wingspan">The wingspan in centimetres</param>
25         public Bird(String name, String diet, String location,
26             double weight, int age, String colour, String species, double wingSpan)
27             : base(name, diet, location, weight, age, colour)
28         {
29             _species = species;
30             _wingSpan = wingSpan;
31         }
32
33         /// <summary>
34         /// Allows the bird to sleep
35         /// </summary>
36         public override void sleep()
37         {
38             Console.WriteLine("{0} lays down and goes to sleep", Name);
39         }
40
41         /// <summary>
42         /// Message posted when the bird tries to fly
43         /// </summary>
44         public virtual void fly()
45         {
46             // code to allow eagles to fly
47             Console.WriteLine("{0} thinks about flying", Name);
48         }
49     }
50 }
```

```
1 using System;
2
3 namespace Task02
4 {
5     /// <summary>
6     /// Prototype for a penguin as a type of bird
7     /// </summary>
8     class Penguin : Bird
9     {
10         /// <summary>
11         /// Constructor for a penguin as a type of bird
12         /// </summary>
13         /// <param name="name">The personal name of the penguin</param>
14         /// <param name="diet">The primary type of food eaten</param>
15         /// <param name="location">The exhibition the penguin is in</param>
16         /// <param name="weight">Weight in pounds</param>
17         /// <param name="age">Age of the penguin in years</param>
18         /// <param name="colour">The dominant color(s)</param>
19         /// <param name="species">The species of bird</param>
20         /// <param name="wingspan">The wingspan in centimetres</param>
21         public Penguin(String name, String diet, String location,
22             double weight, int age, String colour, String species,
23             double wingSpan)
24             : base(name, diet, location, weight, age, colour, species, wingSpan)
25         { }
26
27         /// <summary>
28         /// The penguin lays an egg
29         /// </summary>
30         public void layEgg()
31         {
32             // code to allow penguins to lay eggs
33             Console.WriteLine("{0} lays an egg in the ice.", Name);
34         }
35
36         /// <summary>
37         /// The penguin eats fish
38         /// </summary>
39         public override void eat()
40         {
41             Console.WriteLine("{0} eats 0.5lb of fish", Name);
42         }
43
44         /// <summary>
45         /// The penguin sleeps
46         /// </summary>
47         public override void sleep()
48         {
49             Console.WriteLine("{0} rests in his nest, asleep", Name);
50         }
51
52         /// <summary>
53         /// The penguin goes nowhere in the air
```

```
54     /// </summary>
55     public override void fly()
56     {
57         Console.WriteLine("{0} flaps his little wings and goes nowhere", Name);
58     }
59
60     /// <summary>
61     /// The penguin makes a penguin noise
62     /// </summary>
63     public override void makeNoise()
64     {
65         Console.WriteLine("{0} sneezes", Name);
66     }
67
68     /// <summary>
69     /// The penguin makes it's home
70     /// </summary>
71     public override void buildHome()
72     {
73         Console.WriteLine("{0} builds a rookery", Name);
74     }
75 }
76 }
```

```
1  using System;
2
3  namespace Program_3
4  {
5      class Overloading
6      {
7          public static void methodToBeOverloaded(String name)
8          {
9              Console.WriteLine("Name: " + name);
10         }
11
12         public static void methodToBeOverloaded(String name, int age)
13         {
14             Console.WriteLine("Name: " + name + "\nAge: " + age);
15         }
16
17         static void Main(string[] args)
18         {
19             methodToBeOverloaded("John Doe");
20             methodToBeOverloaded("Jane Doe", 24);
21         }
22     }
23 }
```