Started on	Monday, 28 October 2019, 1:52 PM
State	Finished
Completed on	Monday, 28 October 2019, 2:34 PM
Time taken	42 mins 21 secs

Question **1**

Complete

Marked out of 2.00

For each question, state whether the code is valid or not if placed where the //--line of code goes here--// comment is.
Valid code runs without compile or runtime error.

```
static void Main(string[] args)
{
    Snake mySnake = new Snake();
    Snake myOtherSnake;
     //--line of code goes here--//
}

class Snake
{
   public static string species;
   public string name;
}
```

myOtherSnake.name = "Mr. Slithers";	Valid
name = "Mr.Slithers";	Invalid
mySnake.species = "Cobra";	Invalid
"Mr.Slithers" = Species.Snake;	Invalid
Snake.name = "Mr. Slithers";	Invalid
mySnake.name = "Mr. Slithers";	Valid
Snake.species = "Cobra";	Valid
myOtherSnake.species = "Cobra";	Invalid

Question **2**

Complete

Marked out of 1.00

An error says the length field is inaccessible. How can we fix this problem?

```
static void Main(string[] args)
{
    Snake mySnake = new Snake(3);
    Console.WriteLine(mySnake.length);
}

class Snake
{
     int length;
    public Snake(int _length)
    {
        length = _length;
    }
}
```

Select one:

- a. Set length in Snake to be "public int length;"
- b. Initialize the variable length in snake. "int length = 0;"
- c. declare a variable named length inside the main function.
- d. Make the Snake class public. "public class Snake{...}"
- e. Set length in Snake to be "accessible int length;"

Question **3**

Complete

Marked out of 1.00

Assume a class called Employee that has a single constructor which has the following parameters in order: string name, double hourlySalary.
We want to make a child class called Manager whose constructor has 3 parameters: string name, double hourlySalary, int departmentNumber
We want the Manager constructor to properly pass the two arguments, name and hourlySalary, to the parent constructor.

Which one of the options below is valid?

Select one:

- a. public Manager(string name, double hourlySalary, int departmentNumber) : base(name,hourlySalary) { //..... }
- b. public Manager(string name, double hourlySalary, int departmentNumber) { //..... }
- c. public Manager(int departmentNumber) : Employee(name,hourlySalary) { //..... }
- d. public Manager(string name, double hourlySalary, int departmentNumber) : parent(name,hourlySalary) { //..... }
- e. public Manager(string name, double hourlySalary, int departmentNumber) : Employee(name,hourlySalary) { //..... }
- f. public Manager(int departmentNumber) : Employee(string name, int hourlySalary) { //..... }
- g. public Manager(int departmentNumber) : Employee() { //..... }
- h. public Manager(string name, double hourlySalary, int departmentNumber) : Employee() { //..... }
- i. public Manager(name, hourlySalary, departmentNumber) : Employee() { //..... }

Question **4**

Complete

Marked out of 1.00

```
public class Animal
{
   int numOfLegs;
   string name;
   public Animal(string name, int numOfLegs)
   {
       numOfLegs = numOfLegs;
       name = name;
   }

}

public class Dog : Animal
{
    int numOfLegs;
    string name;
    public Dog(int numOfLegs) : base("Dog", 4) {}
    public void Bark()
    {
        Console.WriteLine(name + " barks");
    }
}
```

Select all the statements which are true about the above code

Select one or more:

- [ ] a. Dog is incorrectly calling the parent constructor; it should be using the Parent keyword instead of base.
- [ ] b. Dog is incorrectly calling the parent constructor, it should be using the Animal keyword instead of base.
- [x] c. Although it does not throw an error, it is a bad idea to have variables with the exact same name in the child, as it hides variables inherited from the parent. In this scenario, we would want to set the variables in Animal to protected, so that they can be accessed from Dog, instead of redefining new ones and breaking our expectations of the functions.
- [ ] d. The Dog constructor does not need to have a numOfLegs parameter. For one thing, this parameter is not used anywhere, and furthermore, we know that dogs have 4 legs.

- [ ] e. The Dog constructor has fewer parameters than the parent constructor which is invalid; Dog is unable to fulfill the argument requirements of the parent Animal constructor.
- [x] f. There is a logical error in the Animal Constructor. Since the constructor declares the variables "name" and "numOfLegs" as parameters, it hides the member variables "name" and "numOfLegs". Therefore, the local parameter variables are being set to equal themselves, rather than the member variables being set to equal the parameters.

Question **5**

Complete

Marked out of 1.00

Below we have the following code

```
static void Main(string[] args)
{
    Shape[] shapes = new Shape[3];
    shapes[0] = new Square(2);
    shapes[1] = new Square(4);
    shapes[2] = new Rectangle(2,3);
    int totalArea = 0;
    for(int i = 0; i < 3; i++)
    {
        totalArea += shapes[i].GetArea();
    }
   Console.WriteLine("totalArea: " + totalArea);
}
abstract class Shape
{
   public virtual int GetArea() { return 0; }
}

class Square : Rectangle
{
   public Square(int _x) : base(_x,_x) {}
}


class Rectangle : Shape
{
   protected int x, y;
   public Rectangle(int _x, int _y)
   {
      x = _x;
      y = _y;
   }
   public override int GetArea() { return x * y; }
}
```

Select all the following options which are TRUE...

Select one or more:

- [ ] a. This is a compiler error because the array can only store instances of Shape, but we are trying to put instances of the children classes Rectangle and Square into it.
- [ ] b. This is an error because arrays can only be filled using object initializer syntax
- [ ] c. There will be an index out of range exception caused by the for loop
- [ ] d. The output of the program will be:
totalArea: 26
- [ ] e. This is a compiler error because the Square constructor passes the same variable twice to the base Shape constructor, which requires no arguments.
- [x] f. This is a runtime error because the Square class does not contain an override for the virtual function GetArea(), so when it is called during the for loop, it will crash with an uninitialized function error.
- [x] g. Shape is an abstract class. Therefore we cannot have an array of Shapes, since we cannot create instances of Shape.
- [ ] h. This is a compiler error because the Square constructor passes the same variable twice to the base Rectangle constructor, which requires two unique arguments.

Question **6**

Complete

Marked out of 0.50

An instance of the class Snake is created in the code below

```
static void Main(string[] args)
{
   Snake mySnake = new Snake();

}

class Snake
{

}
```

Select one:

- (•) True
- ( ) False

Question **7**

Complete

Marked out of 0.50

An instance of the class Snake is created in the code below

```
static void Main(string[] args)
{
   Snake mySnake;
}

class Snake
{

}
```

Select one:

- True
- False

Question **8**

Complete

Marked out of 1.00

Once the code below is run, will each ant have a unique antID?

```
static void Main(string[] args)
{
    Ant[] ants = new Ant[10];
    for(int i = 0; i < 10; i++)
    {
        ants[i] = new Ant();
    }
}

class Ant
{
    public static int numberOfAnts = 0;
    int antID;
    public Ant()
    {
        antID = numberOfAnts++;
    }
}
```

Select one:

- True
- False

Question **9**

Complete

Marked out of 1.00

Once the code below is run, will each salamander have a unique salamanderID?

```
static void Main(string[] args)
{
    Salamander[] salamanders = new Salamander[10];
    for(int i = 0; i < 10; i++)
    {
        salamanders[i] = new Salamander();
    }
}

class Salamander
{
    public int numberOfSalamanders = 0;
    int salamanderID;
    public Salamander()
    {
        salamanderID = numberOfSalamanders ++;
    }
}
```

Select one:

- True
- False

Question **10**

Complete

Marked out of 2.00

Do the following lines of code in the questions get called in the code below? (assuming it was actual code and not comments)

```
static void Main(string[] args)
{
    Ant a = new Ant();
    Bug b = a;
    a.Eat();
    b.Eat();
    b.Sleep();
    a.Gather();
}

class Bug
{
   string name;
   int numOfLegs;
   public Bug(string _name, int _numOfLegs)
   {
      name = _name;
      numOfLegs = _numOfLegs;
   }

   public virtual void Eat()
   {
      //Code A
   }

   public virtual void Sleep()
   {
       //Code B
   }
   public virtual void Gather()
   {
      //Code C
   }

}
```

```
public class Ant : Bug
{
   public Ant() : base(6,"Ant") {}

   public override void Eat()
   {
      //Code D
   }

   public override void Sleep()
   {
       //Code E
   }
   public override void Gather()
   {
      //Code F
      base.Gather();
   }
}
```

Code F	Code is called.
Code A	Code is called.
Code D	Code is called.
Code C	Code is called.
Code B	Code is called.
Code E	Code is not called.

Question **11**

Complete

Marked out of 2.00

Match each piece of code to the statement that applies to it.

```
class SomeClass
{
    public void SomeFunc() { SomeFunc(); }
}
```

This is an example of a compiler error.

```
class SomeClass
{
    public void SomeFunc() {}
    public new void SomeFunc() { int myInt = 5;
}
}
```

This is an example of function overriding.

```
class SomeClass
{
    public void SomeFunc() {}
}
class OtherClass : SomeClass
{
    public new void SomeFunc() {}
}
```

This is an example of function instantiation.

```
class SomeClass
{
    public void SomeFunc() {}
    public void SomeFunc(float someFloat) {}
}
```

This is an example of function overloading

Question **12**

Complete

Marked out of 2.00

Match each piece of code to the statement that applies to it.

`private House MyHouse { get { return value; } }`	Choose...
`public Person Somebody { get; }`	This property has a private get and a private set.
`protected int SomeInt { get; private set; }`	This property can only be read by this class and classes that inherit from this class.
`private float someFloat; public float SomeFloat { get { return SomeFloat; } set { someFloat = value; } }`	This is an example of a compiler error.

Question **13**

Complete

Marked out of 1.00

```
static void Main(string[] args)
{
        int[] arr = new int[] { 4, 2, 3, 4, 5, 6 };
        int sum = 0;
        for (int i = 0; i < arr.Length; i++)
        {
                sum = arr[i];
                i++;
        }
        Console.WriteLine(sum);
}
```

What is the output?

Answer: 5

Question **14**

Complete

Marked out of 1.00

What number will this code output to the console?

```
static void Main(string[] args)
{
    int int1 = 0;
    int int2 = 0;

    while(false || false)
    {
        int1++;
    }

    do
    {
        int2--;
    }
    while(false && true);

    Console.WriteLine(int1 + int2);
}
```

Answer: 0

◄ Weather Delg

Jump to...

Announcements ►