Module 1: Week 2 Review

100% (11/11)

✓ 1. **The default value for a reference type is _______.**

null

✓ 2. **What C# operator is used to create an instance of a class?**

new

✓ 3. **Static methods require an instance of a class to exist in order to be invoked.d**

A True

[✓ B] False

✓ 4. **string s = "Tech Elevator is a coding school";
bool containsWord = s.Contains("Technical");**

What is the value of containsWord at the end of this code segment?

A True

[✓ B] False

✓ 5. **Queue<T> is a FIFO (first in first out) data structure.**

[✓ A] True

B False

✓ 6. **Which collection is an associative collection that supports specific keys assigned to each value?**

A Queue

B Stack

C List

[✓ D] Dictionary

E I don't know

✓ 7. **Dictionary<string, string> zipcodes = new Dictionary<string, string>();**

**zipcodes["44103"] = "Cleveland";
zipcodes["44103"] = "Shaker Heights";**

string city = zipcodes["44103"];

What is the value of city at the end of this code snippet?

"Shaker Heights"

✓ 8. __________ is a principle of object oriented programming that deals with hiding the implementation details of a class to prevent other parties from setting the object with invalid data or into an inconsistent state.

A Inheritance

B Programming

[✓ C] Encapsulation

D Data Protection

E I don't know

✓ 9. Private members are accessible in other parts of the program outside of the class they are defined in.

A True

[✓ B] False

✓ 10. Constructors like methods can be overloaded.

[✓ A] True

B False

✓ 11.

```
//#############################################
// Automobile.cs
//#############################################


public class Automobile
{

private int numberOfPassengers;
public int NumberOfPassengers
{
get { return numberOfPassengers; }
}

private int milesPerGallon;
public int MilesPerGallon
{
get { return milesPerGallon; }
}

private int gasTankCapacityInGallons;
public int GasTankCapacityInGallons
{
```

```
get { return gasTankCapacityInGallons; }
}


public Automobile( )
{
numberOfPassengers = 5;
milesPerGallon = 30;
gasTankCapacityInGallons = 15;
}


public Automobile( int numberOfPassengers, int milesPerGallon, int
gasTankCapacityInGallons )
{
this.numberOfPassengers = numberOfPassengers;
this.milesPerGallon = milesPerGallon;
this.gasTankCapacityInGallons = gasTankCapacityInGallons;
}


public int GetTotalRangeInMiles( )
{
return gasTankCapacityInGallons * milesPerGallon;
}
}


//###########################################
// AutomobileDemo.cs
//###########################################


public class AutomobileDemo
{
public static void main( String[] args )
{
int bestRange;
Automobile genericSedan = new Automobile( );
Automobile miniVan = new Automobile( 7, 25, 20 );

if( genericSedan.getTotalRangeInMiles( ) > miniVan.getTotalRangeInMiles( ) )
{
bestRange = genericSedan.getTotalRangeInMiles( );
}
else

{
bestRange = miniVan.getTotalRangeInMiles( );
}

Console.WriteLine(bestRange);
}
}
```

What number is displayed by the main method of AutomobileDemo?

500