**1) How many constructors can one class have?**

A class can have any number of constructors via overloading.

**2) Why is the Main method specified with the *static* keyword?**

The main method is specified with the static keyword because a static method can be called without instantiating an object. Main needs to be static because in order to allow it to be the entry point in the program.

**3) Which of the simple data types are reference types?**

Of the primitive data types, string is the only reference type. Object is also a reference type

**4) What is the difference between *out* and *ref*?**

There are several differences between *out* and *ref*. They are as follows:

REF:

-The parameter or argument **MUST** be initialized first before it is passed to the *ref*.

-It is **NOT** required to assign or initialize the value of a parameter before returning it to the

Called method.

-Passing a parameter value by ref is useful when the called method is also needed to modify the

pass parameter.

-It is **NOT** mandatory to initialize a parameter value before using it in a called method.

-When we use a ref, data can be passed bi-directionally.

OUT:

-It is not mandatory to initialize a parameter or argument before it is passed as an *out.*

-A called method **IS** required to assign or initialize a value of a parameter before returning to the

called method.

-Declaring a parameter to an *out* method is useful when multiple values need to be returned

from a function or method.

-It **IS** mandatory that a parameter value be initialized within the calling method before its use.

-With *out*, data is passed unidirectionally from the called method to the caller method only.

Additionally, both out and ref are treated differently at runtime, but they are treated the same at compile time.

**5) Write a *for* loop that prints out the odd numbers from 1-11. Use the compound assignment operator “+=” to increment your counter variable.**

static void Main (string[] args)

{

For(int i = 1; i < 12; i += 2)

{

if(i % 2 != 0)

{

Console.WriteLine(i);

}

Console.ReadKey();

}

**6) Write a *while* loop that prints out the odd numbers from 1-11. Use the compound assignment operator “+=” to increment your counter variable.**

static void Main(string[] args)

{

int i = 1;

while (i < 12)

{

if(i % 2 != 0)

{

Console.WriteLine(i);

}

i += 2;

}

Console.ReadKey();

}

**7) Write a *do* loop that prints out the odd numbers from 1-11. Use the compound assignment operator “+=” to increment your counter variable.**

static void Main(string[] args)

{

int i = 1;

do

{

if (i % 2 != 0)

{

Console.WriteLine(i);

}

i += 2;

}

while (i < 12);

Console.ReadKey();

}

**8) Write a snippet of code which:**

**-Calls int.parse(“17”)**

**-handles the appropriate exception if thrown**

**-outputs “done!” to the screen regardless of whether an exception is thrown.**

static void Main(string[] args)

{

string num1 = "17";

try

{

int num2 = int.Parse(num1);

Console.WriteLine(num2);

}

catch (Exception ex)

{

Console.WriteLine(ex.Message);

}

finally

{

Console.WriteLine("Done!");

}

Console.ReadKey();

}