Unit 5 Progress Check: FRQ

method is intended to construct a new `Invitation` object that sets `address` to the value of the parameter and sets `hostName` to the default name `"Host"`. The constructor does not work as intended.

```
public Invitation(String address)
{
       address = address;
       hostName = "Host";
}
```

Write a correct implementation of the one-parameter `Invitation` constructor that avoids the error in the student's implementation.

Write the method below.

```
a) public String getHostName()
   { return hostName;
   }

b) public void updateAddress(String newAddress)
   { this.address = newAddress;
   }

c) public String getMessage(String attendee)
   { return "Dear " + attendee +
   ", please attend my event at " +
   address + ". See you there, " + hostName + ".";
   }

d) public Invitation(String a) {
     address = a;
     hostName = "host" {
```

Unit 5 Progress Check: FRQ

2. SHOW ALL YOUR WORK. REMEMBER THAT PROGRAM SEGMENTS ARE TO BE WRITTEN IN JAVA.

Assume that the classes listed in the Java Quick Reference have been imported where appropriate.
Unless otherwise noted in the question, assume that parameters in method calls are not null and that methods are called only when their preconditions are satisfied.
In writing solutions for each question, you may use any of the accessible methods that are listed in classes defined in that question. Writing significant amounts of code that can be replaced by a call to one of these methods will not receive full credit.

This question involves the implementation of the `PasswordGenerator` class, which generates strings containing initial passwords for online accounts. The `PasswordGenerator` class supports the following functions.

Creating a password consisting of a specified prefix, a period, and a randomly generated numeric portion of specified length
Creating a password consisting of the default prefix `"A"`, a period, and a randomly generated numeric portion of specified length
Reporting how many passwords have been generated

The following table contains a sample code execution sequence and the corresponding results.

```
public class PasswordGenerator{
 private static int pwCount =0;
 private int pwLength ;
 private String prefix ;

 public PasswordGenerator (int len, String pre)
{ pwLength = len ;
  prefix = pre ;
}
 public PasswordGenerator (int len)
{ pwLength = len ;
 prefix = "A" ;
}
```

```
public String pwGen()
{
string temp = prefix + "." ;
for (int i=0; i< pw.length ; i++)
{ temp += (int) (Math.random()*10)
}
pwCount ++ ;
return temp ;
}

}
```