Uncommon Schools | Change History.

AP Computer Science

Q1 Interim Assessment

Section 2: Free Response Questions

School: _____

Student Name: _____

Teacher: _____

Period: _____

Note: Complete these Free Response Questions on these pages. You may use scratch paper, but only work shown on these sheets will be scored!

1. Write a program that simulates the following game: a wheel is divided into 16 sections, numbered from 1 to 16. The game is over when the sum of two consecutive spins is a 20; until then, the value that the player spins gets added to his/her score.

   At the end of the game, your program should print out the player's final score.

   Example
   The following sequence of spins would result in the indicated output:

   Spin #1: 12

   Score:

   Spin #2: 5

   $12 + 5 \neq 20$ → game continues
   Score: 17 (12+5)

   Spin #3: 6

   $5 + 6 \neq 20$ → game continues
   Score: 23 (17+6)

   Spin #4: 14

   $6 + 14 = 20$ → game over

   Final Score: 23

Write your code on the next page!

2. One of the rules for converting English to Pig Latin states: If a word begins with a consonant, move the consonant to the end of the word and add “ay”. Thus “dog” becomes “ogday,” and “crisp” becomes “rispcay”.

   Write a Java class called `PigLatinTranslator`. In it, ask the user to enter an English word that begins with a consonant and store their word in a `String` called `engWord`. Your program should translate `engWord` into Pig Latin and store the translated word in a `String` called `pigWord`. Finally, print out the translated word!

3. Consider the `BobsAmazingCalculator` class below, which Bob plans to use to perform different types of calculations for his friends.

```
public class BobsAmazingCalculator
{
   //@param n: the number to be added to 3
   //@return n plus 3

   public int addThree( int n )
   { /* to be implemented in part (a) */ }


   //@param n: the number to be checked
   //@return true if n is even, false otherwise

   public boolean isEven( int n )
   { /* to be implemented in part (b) */ }


   //@param a, b: the first and last numbers to be added
   //@return: if a < b, return sum of numbers between a and b,
   //inclusive. if not, print "error" and return 0.

   public int findSum( int a, int b )
   { /* to be implemented in part (c) */ }
}
```

   a) Write the `BobsAmazingCalculator` method `addThree` described above. For example, the method call to `addThree(7)` will return the integer `10`.

b) Write the `BobsAmazingCalculator` method `isEven` described above. For example, the method call to `isEven(7)` will return `false`, while the method call to `isEven(4)` will return `true`.

c) Write the `BobsAmazingCalculator` method `findSum` described above. For example, the method call to `findSum(5,7)` will return `18`, while the method call to `findSum(10,4)` will return `0` and result in an "`error`" message.

4. Consider the following incomplete declaration of a `Code` class. Portions of the code may be hidden by changing the corresponding letter or digit to an `X` using the `hide` method. For example, suppose the following `String` object is instantiated:

   ```
   String code = new String("ABCdef123ghi456jklMNO");
   ```

   The following code fragment results in the output indicated.

   Code Fragment
   ```
   String code2 = new String( hide(code, 2, 7) );
   System.out.println( code2 );
   ```

   Output
   ```
   ABXXXXX23ghi456jklMNO
   ```

The `Code` class also has a `wordTriangle` method, which doesn't return anything but takes in a `String` parameter and prints out a "word triangle" based on that word. For example, the method call to `wordTriangle( "triangle" )` would result in the following output:

```
triangle
triangl
triang
trian
tria
tri
tr
t
```

```
public class Code
{
   // precondition: p1 >= 0, p1 < p2
   //            p2 <= myCode.length()
   // Replaces the characters in the code starting at
   // position p1 until p2-1, inclusive, with an X

   public String hide(String code, int p1, int p2)
   {
      // to be implemented in part (a)
   }

   // Prints out a word triangle as described above

   public void wordTriangle(String word)
   {
      // to be implemented in part (b)
   }
}
```

a) Implement the `hide` method.

b) Implement the `wordTriangle` method.