All the questions below pertain to our Assignment 4 and all its code and files. Answer the following questions by selecting the most appropriate and specific answer possible.

1. Which classes in our Assignment 4 are abstract data type (ADT) classes?
   A. Only the data collections classes.
   B. All classes of Assignment 4.
   C. All classes of Assignment 4 except the two exceptions classes.
   D. The BST class and the WordPair class.
   E. None of the above

2. Consider the following C++ method. Fill in the blank with the most appropriate statement such that your selected statement does not create a syntax error and it makes sense in the context of this method.

```cpp
bool WordPair::operator==(const WordPair& rhs) const {
    return (____________________________________) == 0;
} // end of operator==
```

   A. rhs->english.compare(getEnglish())
   B. rhs.english.compare(this.getEnglish())
   C. this.english.compare(rhs->getEnglish())
   D. this->english.compare(rhs.getEnglish())
   E. None of the above

3. Which statement is true about the wrapper methods found in our Assignment 4?
   A. The wrapper methods are all private methods.
   B. Only the BSTNode and BST classes have wrapper methods.
   C. Each wrapper method wraps around a call to a recursive method.
   D. The wrapper methods must throw exceptions.
   E. None of the above

4. The TApp application does not manipulate objects of the BSTNode class. Why?
   A. Because the TApp application’s responsibility is to translate English words, not to maintain a binary search tree (i.e., a data collection) and its nodes.
   B. Because the BST data collection, used by the TApp application, has an array-based data structure, therefore there is no need for the TApp application to manipulate objects of the BSTNode class.
   C. Because the BSTNode class is a template class.
   D. Actually, the TApp application should manipulate objects of the BSTNode class because it is the responsibility of the TApp application to maintain a translator made of BSTNode objects.
   E. None of the above
5. What does the retrieve(...) method of our BST data collection class do when it cannot find the targetElement?

A. It returns false.
B. It returns NULL.
C. It throws the exception ElementAlreadyExistInBSTException.
D. The execution flow returns to the wrapper method without returning any value.
E. None of the above

6. Consider the following C++ method. Fill in the blank with the most appropriate statement such that your selected statement does not create a syntax error and it makes sense in the context of this method.

```cpp
template<class ElementType>
void BST<ElementType>::insert(const ElementType& newElement) throw(ElementAlreadyExistsInBSTException) {
    if (____________________________) {
        root = new BSTNode<ElementType>(newElement);
        elementCount++;
    } else {
        if ( !insertR(newElement, root) )
            throw ElementAlreadyExistsInBSTException("Element already exists in tree.");
    }
    return;
}
```

A. newElement != NULL
B. elementCount > 0
C. elementCount == 0
D. root != NULL
E. None of the above

7. Why is the getter getRoot(...) not offered by the BST class? The description of such a getter is as follows: it returns the value of the pointer root.

A. Because the BST class is an abstract data type (ADT) class.
B. Because the BST class does not have a root data member.
C. Because the root data member of the BST class is public and therefore can be accessed without the use of a getter.
D. Because such getter would allow the client code to access the entire data structure of the BST directly.
E. None of the above
8. Consider the following C++ method. Fill in the blank with the most appropriate statement such that your selected statement does not create a syntax error and it makes sense in the context of this method.

```cpp
template<class ElementType>
bool BST<ElementType>::insertR(const ElementType& anElement, 
    BSTNode<ElementType>* currentRoot) {
    if (anElement == currentRoot->element) return false;

    if ( anElement < currentRoot->element ) {
        if (!currentRoot->hasLeft()) {
            currentRoot->left = new BSTNode<ElementType>(anElement);
            elementCount++;
            return true;
        }
        else
            ____________________________
    }
    else {
        // some more code
    }
}
```

A. insertR(anElement, currentRoot->right);
B. insertL(anElement, currentRoot->left);
C. insertR(currentRoot->left, anElement);
D. insertR(anElement, currentRoot->left);
E. None of the above

9. What does TApp produce on the computer monitor screen when the user enters the command 
`./TApp Display` at the command line?

A. It prints all the pairs of words in alphabetical sorted order based on the English word of the pair then waits for the user to enter an English word to be translated.
B. It prints all the pairs of words in alphabetical sorted order based on the English word of the pair then terminates.
C. It throws an exception.
D. It prints the translation of the English word Display.
E. None of the above
10. What does the parameter of this method represent?

```cpp
template<class ElementType>
void BST<ElementType>::traverseInOrder(void visit(const ElementType&)) const {
    // some code
}
```

A. It indicates that each element stored in the BST is visited as the BST is traversed.
B. It prints the value of the BST’s root on the computer monitor screen.
C. It represents a function defined in the client code TApp.cpp, when this client code calls `traverseInOrder(…)` with that particular function as an argument.
D. It represents a function that modifies its parameter `ElementType`.
E. None of the above

11. Consider the following C++ method. Which statement related to this method is true?

```cpp
template<class ElementType>
int BST<ElementType>::getElementCount() const {
    return elementCount;
}
```

A. The `const` means that the object used to call this method will be the same before and after the method is executed.
B. The `const` means that this method operates in constant time, i.e., in $O(1)$.
C. `return elementCount;` can be replaced by `return this->ElementType;` without affecting the execution of this method.
D. A. and B.
E. None of the above

12. Which statement is true?

A. Duplicated elements are allowed in the TApp application.
B. We really should implement a destructor for the BSTNode class because two of its data members are pointers.
C. The data member `element` in the default constructor of the BSTNode class is not initialised to NULL because the BSTNode is a template class. This signifies that the data type of `element` was unknown when the BSTNode class was implemented.
D. We achieve the time efficiency of $O(\log_2 n)$ for the retrieve and insertion operations of the BST by creating a BST with the shortest possible height. This is done by inserting (into a BST) each element read from the input data file `dataFile.txt` which contains sorted data.
E. None of the above