

# CS215 – Introduction to Program Design, Abstraction and Problem Solving

CS215 Bypass Exam Part I (written exam)(total points: 100) contains:

1. True-or-False questions;
2. Multiple-choice questions;
3. Fill-the-blank questions;
4. Short answer questions;
5. Coding questions.

*The purpose of these practice questions is to give you an idea of the nature of the questions on the exam. They are not meant to cover all the possibilities of questions that could be on the exam.*

## True or False Questions

Circle T for True or F for False, for each of the following questions.

T F (1) A queue is a First-In-First-Out data structure.

T F (2) A linked list is not fixed in size.

T F (3) You can directly access the  $n^{\text{th}}$  node of a linked list.

T F (4) Recursive functions must return a value.

T F (5) When you return a dynamic array to the heap, you must include the number of elements in the array.

## Multiple Choice Questions

**(Only one correct answer to each question)**

1. In the Singly Linked List data structure, each node contains a pointer, which points to
  - a. the data part of a node
  - b. the count part of a node
  - c. the pointer part of the node
  - d. the whole node
2. What line of code assigns a char variable `outputGames` with the value the `gamesPointer` points to?

```
char userGames = 'B';  
char* gamesPointer = &userGames;
```

  - a. `outputGames = gamesPointer;`
  - b. `outputGames = *gamesPointer;`
  - c. `outputGames = &gamesPointer;`
  - d. `gamesPointer = outputGames;`

3. A pointer is a(n) \_\_\_\_\_ that contains a \_\_\_\_\_.
- a. **array, constant value**
  - b. **class, constant value**
  - c. **variable, memory address**
  - d. **constant, memory address**
4. Which of the following lists is sorted?
- a. **Ted, Tom, Thomas, Tim**
  - b. **apple, orange, banana, pear**
  - c. **89, 79, 63, 22**
  - d. **1.01, 1.02, 1.1, 1.001**
5. Which constructor will be called by the statement **House myHouse (12) ;** for the given code?
- ```
class House {  
    House(); // Constructor A  
    House(int num, string str); // Constructor B  
    House(int num1, int num2); // Constructor C  
    House(int num); // Constructor D  
};
```
- a. **Constructor A**
  - b. **Constructor B**
  - c. **Constructor C**
  - d. **Constructor D**
6. Which of the following statements is true about private data members of a class?
- a. **A class user can access the private data members to operate on the function.**
  - b. **A class user and member functions can access the private data members to operate on the function.**
  - c. **A member function can access the private data members, but class users cannot.**
  - d. **A member function cannot access the private data members to operate on the function.**
7. What is the return type for constructor?
- a. **int**
  - b. **char**
  - c. **void**
  - d. **none**
8. Which of the following statements is true about a memory leak?
- a. **A memory leak is a memory location that has been dynamically allocated but no longer used by a program.**
  - b. **A memory leak occurs when a program allocates memory but loses the ability to access the allocated memory.**
  - c. **A memory leak is an automatic process of finding and freeing unreachable allocated memory locations.**
  - d. **A memory leak is a process used to determine if the memory is usable or unusable.**

9. Which XXX completes the following code, assuming a class called Movies has been defined with public member functions `GetDetails()` and `SetDetails()`?

```
int main() {
    double movieRating;
    string actors;
    XXX
    cout << "The movie's rating and its actors are: " <<
newMovie.getDetails() << endl;
    return 0;
}
```

- a. `Movies myMovie;`  
    `myMovie.setDetails();`
- b. `Movies myMovie;`  
    `myMovie.GetDetails();`
- c. `Movies newMovie;`  
    `newMovie.GetDetails();`
- d. `Movies newMovie;`  
    `newMovie.SetDetails();`

10. Which XXX completes the binary search algorithm?

```
int BinarySearch(int numbers, int numbersSize, int key) {
    int mid = 0;
    int low = 0;
    int high = numbersSize - 1;

    XXX {
        mid = (high + low) / 2;
        if (numbers[mid] < key) {
            low = mid + 1;
        }
        else if (numbers[mid] > key) {
            high = mid - 1;
        }
        else {
            return mid;
        }
    }
    return -1;
}
```

- a. `while (high >= low)`
- b. `if (high >= low)`
- c. `while (high <= low)`
- d. `if (high <= low)`

11. Which of the following lists represent a possible inorder scan of a binary search tree?

- a. 7 3 8 2 9 4 11
- b. 2 3 4 7 8 9 11
- c. 11 2 9 3 8 4 7
- d. **All of the above.**

## Fill the blank questions

1. Show the output of the following code:

```
void swap(int a[], int size)
{
    if (size < 2) return;
    int temp = a[ 0 ];
    a[ 0 ] = a[ 1 ];
    a[ 1 ] = temp;
}
int main()
{
    int a[] = {10, 20};
    swap(a,2);
    cout << "Line 1: " << a[ 0 ] << endl;
    cout << "Line 2: " << a[ 1 ] << endl;
    cout << "Line 3: " << *(a+1) << endl;
    cout << "Line 4: " << *a+1 << endl;
    return 0;
}
```

The output is:

Line 1: \_\_\_\_\_

Line 2: \_\_\_\_\_

Line 3: \_\_\_\_\_

Line 4: \_\_\_\_\_

## Coding Questions

*Provide complete definition for each function: remember to declare all variables. You do not have to write a **main** function or include header files.*

1. Given the following structure for a binary tree node:

```
class Node {
public:
    Node(int new_value);
    int value;
    Node *left;
    Node *right;
};
```

And the following (partial) definition of a binary tree class:

```
class Tree {
public:
    Tree(); // Create a new empty tree.
    int size() const; // return the number of nodes
private:
    Node *root;
};
```

- a. Implement the constructor for the **Tree** class.
- b. Define a recursive function **subtree\_size** that takes a pointer to a **Node** and returns the size of the subtree beneath that node. Remember that the null pointer represents an empty subtree.
- c. Implement the **size** member function of the **Tree** class using **subtree\_size**.