**A506 / C201 Computer Programming II**
**Placement Exam Sample Questions**

For each of the following, choose the most appropriate answer (2pts each).

_____ **1**. Which of the following functions in a class Date is a constructor?

```
class Date {...
```

a. `Date(int m, int d, int y);`
b. `void Add(Date from);`
c. `void copyDate(Date from);`
d. All of the above

_____ **2**. A method is overloaded in the derived class when
a. a method from the base class is called on an object of the derived class;
b. the derived class implements a method with the same name and parameters as the base class
c. the base class implements two methods with the same name
d. none of the above

_____ **3**. Which of the following applies to a class derived from more than one class?
a. multiple inheritance
b. the Java compiler does not accept that
c. simple inheritance
d. polymorphism

_____ **4**. What does polymorphism mean in Java and other languages?
a. Two functions with the same name and the same parameters.
b. Functions returning more than one value.
c. Two functions with the same name but different sets of parameters.
d. Two classes with the same name.

_____ **5**. Which of the following is *false* about static attributes?
a. They can be accessed from static methods.
b. They have a single instance for the whole class.
c. They can be accessed from non-static methods.
d. Each object of the class has its own version of them.

**6.** Write a static function that will accept a float array and then return the average of the numbers and the first number in the array. The function should use an array to return the values. Also write the part of the main function that shows how the function is called, and then the results retrieved into the two variables.

```
static void main()
{
    float average, first;
    float [] N = {5,6,8,3,6};


}
```

**7.** Draw the linked structure that results from calling the function testList below. Do not include temp in your final drawing.

```
class Node
{
   int datum;
   Node link;

    static void testList()
    {
        Node L, temp;
        L = new Node();
        L.datum = 2;
        L.link = new Node();
        L.link.datum = 3;
        temp = new Node();
        temp.datum = 1;
        L.link.link = temp;
        temp.link = L;
    }
}
```

L

**8.** In the same class, implement the following function:

```
// Finds the last node and returns a reference to it.
// If the list is empty, it returns null.
static Node last(Node front)
{
```

**9.** Write a program that accepts command line arguments. If there are no arguments, then output 0. If there is one argument, assuming that it is an integer, output its square. We will assume that the argument will be a character in the range '0' to '9'.

**10**. What happens when you try to call a class method from a variable that has not been assigned a new object of the class?

**11**. Assume that the array "a" was declared the following way. Write a set of instructions that outputs all the elements, using the for-each loop.

```
float [] a = new float[10];
```

**12**. In the following recursive function, identify the base case and the recursive call.

```
static int factorial(int n)
{
    if (n < 2)
       return 1;
    else
        return n * factorial(n-1);
}
```

**13.** Complete the following class, so that the program can display the required results.

Note: You need to allocate space if needed.

```
class Book
{
    String title;           //every book has a title
    int editions;
    double [] price;        //every book has a price

// Constructor of Book, assign parameters aTitle and aPrice to Book
// attributes title and price, respectively. Single edition.
public Book(String aTitle, double aPrice)
{
     // Your code goes here


}


// Constructor of Book with multiple editions with several prices.
public Book(String aTitle, double [] prices)
{
    // Your code goes here

}
public String getTitle()
{
    return title;
}
```

```
public double getPrice(int which)
{
    if (0 <= which && which < editions)
        return price[which];
    else
        System.exit(1); // quit the program with an error message
}

public double getLowestPrice()
{
    // Your code goes here


}
```

**14.** Write a static function that takes a string as parameter and counts the digits in it, which are all the characters between '0' and '9'.

**15**. Write a static function that takes an array of integers as parameter and an integer value, and returns the index of the first occurrence of the value in the array, or -1 if the value is not there.

## Answers

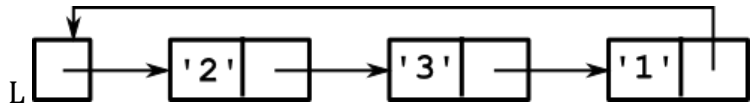
1. a
2. b
3. b
4. c
5. d

**6.**

```
static void main()
{
    float average, first;
    float [] result;
    float [] N = {5, 6, 8, 3, 6};
    result = ComputeAverage(N);
    average = result[0];
    first = result[1];
}

float [] ComputeAverage(int a[])
{
    float [] result = new float[2]
    float sum = 0;
    for (int i = 0; i < a.length; i++)
        sum += a[i];
    result[0] = sum/MAX;   // average
    result[1] = a[0];      // first
    return result;
}
```

**7.**




**8.**

```
static Node last(Node front)
{
    if (front == NULL)
        return NULL;
    Node lookup = front;
    while (lookup.link)
        lookup = lookup.link;
    return lookup;
}
```

**9.**

```
static void main(String [] args)
{
    if (args.length == 1)
        System.out.println(0);
    else {
        int n = args[1][0]-'0';
        System.out.println(n*n);
    }
}
```

**10.** You get a runtime error mentioning a void pointer exception.

**11.**

```
for (float x: a)
        System.out.println(x);
```

**12.**

```
long factorial(int n)
{
   if (n < 2)  // ← base case
    return 1;
   else
      return n * factorial(n-1);
}                // ^recursive call
```

**13.**

```
Book(String aTitle, double aPrice)
{
     title = aTitle;
     editions = 1;
     price = new double[1];
     price[0] = aPrice;
}

Book(String aTitle, double [] prices)
{
    int count = prices.length;
    title = aTitle;
    if (count > 0) {
        editions = count;
        price = new double[count];
        for (int i = 0; i < count; i++)
            price[i] = prices[i];
    }
}
```

```
double getLowestPrice()
{
    if (editions == 0)
        return 0.0;
    double minPrice = price[0];
    for (int i = 1; i < editions; i++)
        if (price[i] < minPrice)
            minPrice = price[i];
    return minPrice;
}
```

14.

```
static int countSpaces(String text)
{
    int count = 0;
    for (int i = 0; i < text.length; i++)
        if (text[i] == 0)
            count++;
    return count;
}
```

15.

```
static int firstIndex(int [] a, int val)
{
    for (int i = 0; i < a.length; i++)
        if (a[i] == val)
            return i;
    return -1;
}
```