CSCE 146 Practice Midterm 1 Answers

This exam totals 100 points. Read each question carefully and follow the instructions. Each question has one or more correct answers. Make sure to mark all correct answers. Good luck!

00 Programming Review

1. Based on the following code snippet, clearly fill in ALL true statements.

Code Snippet

```
final int a = 10;
for(int i=0;i<a;i+=2)
{
    for(int j=i;j<a;j++)
    {
        System.out.print("*");
    }
    System.out.println();
}
```

   - ○ The first line from the top will print 9 asterisks (“*”).
   - ○ The last line from the top will print 9 asterisks (“*”).
   - ○ The first line from the top will print 2 asterisk (“*”).
   - ○ The last line from the top will print 2 asterisks (“*”).
   - ○ All the above statements are false.

2. Based on the following code snippet, clearly fill in ALL true statements.

Code Snippet

```
int[] a = {1,2,3,4,5,6,7,8,9,10};
for(int i=1;i<5;i++)
{
    System.out.print(a[i]);
    System.out.println(" "+a[i*2]);
}
```

   - ○ The second line will print “3 5”.
   - ○ The second line will print “3 6”
   - ○ The third line will print “4 7”
   - ○ This loop will cause an “Index Out of Bounds Exception”.
   - ○ All the above statements are false.

01 Programming Review – File I/O

3. Based on the following code snippet (above) and file (below), clearly fill in ALL true statements.

`public static int countLetter(String fileName, char letter)`
`{`
`  int count = 0;`
`  try`
`  {`
`  Scanner fileScanner = new Scanner(new File(fileName));`
`  String next = "";`
`  while(fileScanner.hasNext())`
`  {`
`  next = fileScanner.next();`
`  for(int i=0;i<next.length();i++)`
`  {`
`  if(next.charAt(i)==letter)`
`  count++;`
`  }`
`  }`
`  fileScanner.close();`
`  }`
`  catch(Exception e)`
`  {`
`  e.printStackTrace();`
`  }`
`  return count;`
`}`
	Code Snippet

I met a traveller from an antique land\n Who said: Two vast and trunkless legs of stone\n Stand in the desart.\n	File

- Provided with the file and the letter 'i', the method will return 3.
- Provided with the file and the letter 'i', the method will return 4.
- Provided with the file and the letter 'e' the method will return 9.
- Provided with the file and the letter 'E' the method will return 0.
- All the above statements are false.

4. Based on the following code snippet (above) and file (below), clearly fill in ALL true statements.

``
public static int formatCheck(String fileName)
{
    int errors = 0;
    try
    {
        Scanner fileScanner = new Scanner(new File(fileName));
        String fileLine = "";
        String[] splitLine;
        while(fileScanner.hasNextLine())
        {
            fileLine = fileScanner.nextLine();
            splitLine = fileLine.split("\t");
            if(splitLine.length != 3)
                errors++;
        }
        fileScanner.close();
    }
    catch(Exception e)
    {
        e.printStackTrace();
    }
    return errors;
}
``
	Code Snippet

a\tb\tc\n 1\t2\t3\n e f g\n 5\t6\n 7\n h\ti\tj\n 8\t9\t10\n the\tquick\tbrown\n fox\tjumps\tover the lazy\tdog\n	File

- The method will return 2
- The method will return 3
- The method will return 4
- The method will return 5
- All the above statements are false.

02 Linked Lists

5. Based on the following code snippet, clearly fill in ALL true statements.

Code Snippet

```
private class ListNode
{
    String data;
    ListNode link;
    public ListNode(String aData, ListNode aLink)
    {
        data = aData;
        link = aLink;
    }
}
public void linkedTest()
{
    ListNode head = new ListNode("z",null);
    head.link = new ListNode("y",null);
    head.link = new ListNode("x",head.link);
    head.link.link = new ListNode("w",null);
    head.link.link = new ListNode("y",head.link.link);

    for(ListNode temp = head;temp!=null;temp = temp.link)
        System.out.println(temp.data);

}
```

- ○ The second line printed is “y”
- ○ The second line printed is “x”
- ○ The third line printed is “y”
- ○ The third line printed is “w”
- ○ All the above statements are false.

6. Based on the following code snippet, clearly fill in ALL true statements.

Code Snippet

```
private class ListNode
{
    String data;
    ListNode link;
    public ListNode(String aData, ListNode aLink)
    {
        data = aData;
        link = aLink;
    }
}
public void linkedTest02()
{
    ListNode head = new ListNode("a",null);
    head = new ListNode("b",head);
    head = new ListNode("c",head);
    head = new ListNode("d",head);
    head.link.link = null;
    head = new ListNode("e",head);
    head = new ListNode("f",head);

    for(ListNode temp = head;temp!=null;temp = temp.link)
        System.out.println(temp.data);
}
```

- The first line will print “f”
- The last line will print “c”
- The last line will print “a”
- The next to the last line will print “b”
- All the above statements are false.

03 Queues

7. Based on the following code snippet, clearly fill in ALL true statements.

Code Snippet

```
Queue<Integer> intQueue = new LinkedList<Integer>();
for(int i=0;i<21;i+=3)
{
    intQueue.add(i);
}
while(intQueue.size()>3)
{
    intQueue.remove();
}
for(int i : intQueue)
{
    System.out.println(i);
}
```

- ○ The first line printed will be 0
- ○ The second line printed will be 15
- ○ The third line printed will be 18
- ○ The fourth line printed will be 12
- ○ All the above statements are false.

8. Based on the following provided Queue implemented as a circular array and pseudocode, clearly fill in ALL true statements.

Initial Queue

Index	0	1	2	3	4	5	6	7
Value	25	74	84	3	23	85	96	-

The head and tail values are at index 0 and index 6 respectively.

Pseudocode

1. Dequeue 3 times
2. Enqueue 7
3. Enqueue 12
4. Enqueue 23
5. Dequeue 2 times

- ○ The head index will be at 5
- ○ The tail value will be 23
- ○ The head index is always at 0
- ○ The tail index is always at 7
- ○ All the above statements are false.

04 Stacks

9. Based on the following code snippet, clearly fill in ALL true statements.

Code Snippet

```
Stack<Double> dubStack = new Stack<Double>();
for(double i=0;i<100.0;i+=13.0)
{
    dubStack.push(i);
}
for(int i=0;i<2;i++)
{
    dubStack.pop();
}
dubStack.push(50.0);
while(dubStack.isEmpty()==false)
{
    System.out.println(dubStack.pop());
}
```

- ○ The first value printed will be 0.0
- ○ The first value printed will be 50.0
- ○ The last value printed will be 0.0
- ○ The second value printed will be 65
- ○ All the above statements are false.

10. Based on the following provided Stack implemented as an array and pseudocode, clearly fill in ALL true statements.

Initial Queue

Index	0	1	2	3	4	5	6	7
Value	25	74	84	3	23	85	96	-

The head value is at index 6.

Pseudocode

1. Push 23
2. Pop 3 times
3. Push 18
4. Push 22
5. Pop

- ○ The value at index 0 will be 23
- ○ The value at index 1 will be 74
- ○ The head value will be 18
- ○ The head value will be 22
- ○ All the above statements are false.