

Exam1 Covers Appendix A, B, C, D, Chapters 1,5,10,12 (plus linked nodes and array review)

Java Review:

- 1) (Primitive variables, read) what does the following code print?

```
double x = 3.5;
int y = 5;
char z = 'q';
boolean w = false;

x = x + y;
y = y - 1;
w = (z == 'q');
System.out.println("x = " + x + ", y = " + y + ", z = " + z + ", w = " + w);
```

**x = 8.5, y = 4, z = q, w = true**

---

- 2) (Primitive variables, write) write code that **computes** and **prints** the correct miles per gallon (as a floating point value) given the data stored in the variables. (can be done in one line). Don't use numbers, only variables, to compute mpg.

```
int gallons = 7;
int miles = 160;
double mpg = (double)miles/gallon;

System.out.println("MPG = " + mpg);
```

---

\_\_\_\_\_ ; // print "MPG = " then mpg

- 3) (Loops and Branches, read) What will the following loop print to the screen?

```
int count = 7;
while ( count >= 1 )
{
    if ( count < 5 )
        System.out.print( count );
    else
        System.out.print( "X" );
    count = count - 1;
}
```

OUTPUT: \_\_\_\_\_ **XXX4321** \_\_\_\_\_

- 4) (Loops and Branches, write) Fill in the blanks for this program, which computes the sum of the numbers entered that are within the range of 20 to 50 inclusive. The user stops the program by entering a value of 0.

For example, if the user enters the values 24, 55, 34, 12, 0 the program prints "the inrange sum is 58" (i.e. 24 + 34)

```
int sum = 0;
Scanner keyboard = new Scanner(System.in);
int number;

do
{
    System.out.println("Enter a number, 0 to finish");
    number = keyboard.nextInt();

    if ( number >= 20 && (number <=50) )
        sum += number;
} while( number != 0 );

System.out.println("The inrange sum is " + sum);
```

- 5) (Static methods, read) the following static method **fun** is defined as follows:

```
public static int fun(int x)
{
    x--;
    return x + 2;
}
```

What is the value of **x** after the following code segment is executed?

```
int x = 3;
x = fun(x);
System.out.println("x = " + x);           x = 4
```

- 6) (Static methods, write) Write a static method that accepts a chest size in inches and returns a char letter size ('L', 'M', or 'S'); Assume large is 42 or more, small is less than 36, and medium is in between. For example, `letterSize(40)` will return the value 'M'

```
public static char letterSize( int inches )
{
    if (inches >= 42)
        return 'L';
    else if (inches >= 36)
        return 'M';
    else return 'S';
}
```

7) (String variables, read) What does the following code print?

```
String word = "braggadocio";
System.out.println( word.charAt(0) );

System.out.println( word.charAt(word.length() - 1));

System.out.println( word.equals("braggadoc"));

System.out.println( word.compareTo("broadway")<0);
```

OUTPUT

    **b**    

    **o**    

    **false**    

    **true**    

8) (String variables, write) If the word is between "alpha" and "bravo" print it out

Write some code that prints out String word, only if it comes in the dictionary between "alpha" and "bravo"

```
if (word.compareTo("alpha")>0 && word.compareTo("bravo") < 0)
    System.out.println(word + " comes between alpha and bravo");
```

9) (Basic class methods, read) Consider the following class:

```
public class Widget
{
    private int nuts, bolts;

    public Widget(int nuts, int bolts) { this.nuts = nuts; this.bolts = bolts; }

    public int getBolts() { return bolts; }

    public boolean equals(Object other) { return bolts == ((Widget)other).bolts; }

    public String toString() { return "N:" + nuts + ", B:" + bolts; }
}
```

What will be printed when the following statements execute in a main method:

OUTPUT

```
Widget w1 = new Widget(5,4);
Widget w2 = new Widget(8,4);

System.out.println("Widget w1 = " + w1);

System.out.println("w1 equals w2? " + w1.equals(w2));
```

**Widget w1 = N:5, B:4**

**w1 equals w2? true**

10)(Basic class methods, write)

Write a setBolts method for the Widget class:

```
public void setBolts(int bolts)
{
    this.bolts = bolts;
}
```

If nuts cost 0.10 dollars and bolts cost 0.15 dollars, write a method calculatePrice for the Widget class, that returns how much a particular Widget object costs.

```
public double calculatePrice()
{
    return 0.1*nuts + 0.15*bolts;
}
```

11)(Array of primitive, read) Trace this program and list the values in the scores array, after it runs, separated by commas. Don't use Java, use paper!

```
int [] scores = {12,23,34,45,56,67,78,89,90};

for (int k = 0; k < 5 ; k++) {
    scores[k] = scores[k+4];
}

56, 67, 78, 89, 90, 67, 78, 89, 90
```

12)(Array of primitive, write) Assign the scores array the following sequence of numbers using a math formula in a for loop (start with k):

50, 150, 250, 350, 450, 550, 650, 750, 850

```
for (int k=0; k < scores.length; k++)
    scores[k] = k*100 + 50;
```

13)(Array of objects, read) What will be printed by the following code?

```
Item [] returns = new Item[5];

returns[0] = new Item("Shoes",79.99);
returns[1] = new Item("Coat", 64.99);

for (int k=2; k< returns.length; k++)
    returns[k] = returns[k-2];

returns[0].setPrice(49.99);

System.out.println( returns[3] + ", " + returns[4]);

"Coat", 64.99, "Shoes", 49.99
```

---

14) (Bags, read) What's in the bag words2 after these statements execute? Show the words printed by the following code.

```
Bag<String> words = new Bag<String>();
Bag<String> words2 = new Bag<String>();

words.add("aardvark");
words.add("breeze");
words.add("acrimony");
words.add("advantage");
words.add("dentist");

while(!words.isEmpty())
{
    String temp = words.remove();
    if (temp.charAt(0)=='a')
        words2.add(temp);
}

System.out.println(words2);
```

**aardvark, acrimony, advantage [in any order - order is not specified for Bag]**

15) (Bags, write) Assume groceryBag is a Bag of Item objects. Empty out and determine the total cost of all the items in groceryBag. Use the getPrice method of the Item class

```
double total = 0.0;

while(!groceryBag.isEmpty())

    total += groceryBag.remove().getPrice();
```

16)(Stacks, read) What's in the stack r after the following statements execute? List the items in the stack r from bottom to top, separated by commas:

```
StackInteface<Integer> s = new ArrayStack<Integer>();
for(int k=0; k<50; k+=10)
    s.push(k);

StackInteface<Integer> r = new ArrayStack<Integer>();
while(!s.isEmpty())
    r.push(s.pop());
```

**40 30 20 10 0**

---

17)(Stacks, read) Evaluate the following postfix expressions given that  
a = 2, b = 3, c = 4, d = 5, e = 6

a) a b + c d - \*                      -5  

b) a b c \* + d -                        9

18)(Stacks, write) Fill in the blank to remove all the items from Stack s until the String "stop" is reached. The String "stop" remains on the stack and is never removed.

```
while( !s.peek().equals("stop") )  
    s.pop();
```

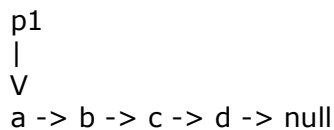
19)(Linked nodes, read) What is printed by the following?

```
Node p1 = new Node("a",null);  
Node p2 = new Node("b",null);  
Node p3 = new Node("c",null);
```

a) p1.next = p2;  
System.out.println(p1.data + " " + p1.next.data);     **a b**    

b) p1 = p2;  
System.out.println(p1.data + " " + p2.data);     **b b**    

20)(Linked nodes, write) Given the following list configuration



Write a single java statement that moves p1 so it refers to the "c" node

```
p1 = p1.next.next;
```

21)(Queues, read) After the following statements execute, what are the contents of the queue? (list items in order, with front item at left)

```
QueueInterface<String> myQueue = new LinkedQueue<String>();  
myQueue.enqueue("Jane");  
myQueue.enqueue("Jess");  
myQueue.enqueue("Jill");  
myQueue.enqueue(myQueue.dequeue());  
myQueue.enqueue(myQueue.getFront());  
myQueue.enqueue("Jim");  
String name = myQueue.dequeue();  
myQueue.enqueue(myQueue.getFront());
```

**Jill, Jane, Jess, Jim, Jill**

22)(Queues, write) Suppose myQueue is a Queue of <String>. Write a statement that checks if the String in front equals "aloha", and if it does, remove it from the Queue.

```
if (myQueue.getFront().equals("aloha"))  
    myQueue.dequeue();
```

23)(Lists, read) After the following statements execute, what is the order of the contents of the list? (in order, with position 1 at left)

```
LList<String> myList = new LList<String>();  
myList.add("alpha");  
myList.add(2, "beta");  
myList.add("gamma");  
myList.add(1, "delta");  
myList.add(3, "alpha");  
myList.remove(1);  
myList.remove(3);  
myList.replace(2, "delta");
```

```
alpha, delta, gamma
```

24)(Lists, write) Suppose quizScores is a LList<Double> filled with quiz scores. Write Java statements that will compute the average of the scores remaining in the list.

```
double avg = 0.0;  
for (int k=1; k<quizScores.getLength() )  
{  
    avg += quizScores.getEntry(k) ;  
}  
avg = avg/quizScores.getLength();
```