PRINT YOUR NAME:     KEY

I have not looked at anyone else’s paper, and I have not obtained unauthorized help in completing this exam. Also, I have adhered to and upheld all standards of honesty as stated in the University Honesty Policy and in the course syllabus.

YOU ARE ALLOWED TO HAVE ONLY THE FOLLOWING ON YOUR DESK OR WORKTABLE:

1. A HANDWRITTEN CRIB SHEET 8-1/2 x 11 inches (ONE SIDE ONLY)
2. A BASIC CALCULATOR – NO GRAPHING, NO PROGRAMMABILITY
3. YOUR PEN OR PENCIL AND AN ERASER

ALL OTHER OF YOUR BELONGINGS BROUGHT INTO THE CLASSROOM MUST BE ON THE FLOOR, UNLESS YOU REQUIRE EYEGlasses OR AN OPTICAL MAGNIFYING GLASS.

PLEASE TURN ALL CELL PHONES, iPHONES, iPADS, AND OTHER ELECTRONIC DEVICES (EXCEPT CALCULATORS), OFF AND PUT THEM AWAY – RINGTONES AND CELL PHONE CONVERSATIONS NOT PERMITTED DURING EXAM.
This exam has four regular questions and one extra-credit question. Complete the questions that are easiest for you first, then complete what you can of the difficult questions. There is no penalty for guessing. However, on questions involving calculation, you must show your work. If you do not show your work, you risk getting only partial credit for any answer.

Q1. (25 pts) Define the following terms using 1-3 sentences or a formula (5 pts each):

(a) **Compilation** – In Java, compilation is the process of converting Java code (.java files) into byte code (.class files). The Java compiler is javac. In general, compilation converts a higher level language into a lower level language.

**Rubric:** 5pts for knowing compilation goes from (java code/.java files/source code/high level language/etc) to (byte code/.class files/object code/low level language/etc). 3pts for mentioning javac but not properly explaining what it does.

(b) **Floating Point Number** – a data type that is capable of holding numerical data with a fractional component. It is distinct from fixed point representation because the number of bits used for representing the fractional and integer components is not fixed.

**Rubric:** 2pts for knowing it holds fractional data, 3 points for explaining that the position of the decimal point can “float” (ie. is not fixed)

(c) **Postcondition Loop** – a loop is a block of code that repeats its body until its condition becomes false. A post condition loop checks its condition for the first time after executing the loop once.

**Rubric:** 2pts for defining loop, 3pts for explaining what “post-condition” means

(d) **Object** – an instance of a class.

**Rubric:** full credit given for any answer that understands an object is a class with state information. 3pts given for defining object logically (OOP concept).

(e) **Class** - specifies a blueprint for objects. A class describes the data that can be stored in an object (variables) and behavior of the object (methods).

**Rubric:** full credit for knowing class specifies data (variables) AND behavior (methods). 3pts for just explaining that it is a blueprint for objects, or only identifying one.
Q2. (35 pts) Given the following PSEUDO CODE, answer each of the following two questions completely (15 pts for 2.1, 20 pts for 2.2):

```plaintext
procedure looper( )
    int M, j = 0, N = 4; float x = 1.1; M = N + 1;
    float outarray[M]
    while j <= N do:
        x = 2 * x + j; j++ ; outarray[j] = x + 3
    
2.1) What are the values of the following variables after each pass through the loop? If at any point an error condition occurs, please state the error clearly. (show in table):

<table>
<thead>
<tr>
<th>j = ?</th>
<th>x = ?</th>
<th>outarray[j] = ?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Before 1st Pass</td>
<td>0</td>
<td>1.1</td>
</tr>
<tr>
<td>Start After 1st Pass</td>
<td>1</td>
<td>2.2</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>5.4</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>12.8</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>28.6</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

2.2) Write the JAVA CODE FRAGMENT for the pseudocode shown above using a for loop to generate the same result as shown above (comment your code to get full credit):

```java
//declare and initialize variables M,N,x
int M, N = 4;
float x = 1.1f;
M = N + 1;

//create an array
float outarray[] = new float[M];

//precondition loop repeats until j > N
for(int j=0; j <= N; j++){
    x = 2 * x + j;
    outarray[j+1] = x + 3; //note the j+1 because we
} //changed order of j++ operation
```
PRINT YOUR NAME:  KEY

Q3. (25 pts) Circle the errors in the Java code shown below, and correct each error by writing the correct symbol or expression in the spaces provided next to each statement or describing the correction if appropriate. IF THERE ARE NO ERRORS IN A LINE OF CODE, LEAVE THE SPACE BLANK.

4 free points here for attempting the problem. -2pts for each error you identified that was not an error.

```java
public static float In2Cm(float inches) ;  // static float to class, no input args, missing {.
    float cm, inches = 0;                  // also accepted method missing body (3pts)
    cm = inches * 2.54f;                    // can’t store a double in a float (3pts)
    return inches;                         // no error, should be left blank (3pts). *See above
}                                           // inches should be cm, but no penalty for missing that.
```

Q4. (35 pts) Write Java code for a class Looper2 with a method Odd to do the following:

1. In a loop, increase a variable “j” from 1 to 9 (inclusive) in steps of 2 (e.g., 1, 3, 5 …)
2. Within the loop, add a letter to a string “s” that is initialized to an empty string – the letter you add should be “A” for j = 1, “B” for j = 2, and so forth.
3. The last statement within the loop displays string “s” using System.out. ...

   COMMENT YOUR CODE FULLY TO GET FULL CREDIT!!

```java
public class Looper2{
    public static void Odd(){
        //initialize empty string
        String s = " ";
        //i=[1,9] with step size 2
        for(int i=1; i <= 9; i += 2){
            //convert 123456789 to ABCDEFGHI
            //skipping BDFH since i steps by 2
            if(i == 1) s = s + "A";
            else if(i == 3) s = s + "C";
            else if(i == 5) s = s + "E";
            else if(i == 7) s = s + "G";
            else if(i == 9) s = s + "I";
            //print s
            System.out.println(s);
        }
    }
}
```
Extra Credit Problem:

EC. (20 pts) The Fibonacci numbers are 1, 1, 2, 3, 5, 8, 13, … that is, the $i^{th}$ Fibonacci number $F(i)$ is the sum of the two numbers that preceded it, namely, the Fibonacci numbers $F(i-1)$ and $F(i-2)$. Given that $F(0) = 0$ and $F(1) = 1$, write a method in Java called `MakeFib` with a loop and an array to generate and store the Fibonacci numbers from $i = 0$ to $i = 10$ (inclusive). PLEASE USE A LOOP (iteration), NOT RECURSION.

PLEASE COMMENT YOUR CODE SO WE KNOW WHAT IT IS SUPPOSED TO DO (and so we can give full credit for a correct result).

```java
class MakeFib {
    public static void MakeFib()
    {
        // create an array to hold the first 11 Fibonacci numbers
        int[] fibNums = new int[11];

        // walk over each number
        for(int i=0; i < fibNums.length; i++){
            // F(0) defined to be 0
            if(i == 0){
                fibNums[i] = 0;
            } // F(1) defined to be 1
            else if(i == 1){
                fibNums[i] = 1;
            } // F(i) = F(i-1) + F(i-2) for i >= 2
            else{
                fibNums[i] = fibNums[i-1] + fibNums[i-2];
            }
        }
    }
}
```