

Name:

**Getting Started**

- Please type your name into the box provided at the top, save the PDF, close the file, and reopen it. Your name should still appear there. If it does not, or if you are unable to enter values into the form field, contact your Instructor right away!
- Once you are able to enter data into the form fields and save the PDF, please rename this file as *FirstnameLastname.pdf*, using your *Firstname* and *Lastname*.

**Exam Instructions**

- This exam consists of four parts and ends on page 9. Page 10 is provided in case you need additional space for any of your answers. You can leave it blank if not needed.
- The exam should take 80 minutes to complete but you must submit by the deadline as specified by your instructor.
- To submit, be sure you have named the file as *FirstnameLastname.pdf* and email it to your instructor by the specified deadline; your instructor will send you a confirmation email once it is received.
- All resources (textbook, class notes, completed labs and assignments, etc.) are permitted, but you may not receive assistance from another person.
- It is okay to email the instructor if you have a clarification question; please do not ask a TA or post a note on Piazza
- There will be no partial credit for incorrect use of Java syntax. You may use your code editor, e.g. VS Code, to confirm the correct syntax.
- When writing Java code inside loops or if-statements, use the spacebar to indent your code in case the Tab key does not work.

Sign the following statement **after** you have completed the examination by typing your name in the box provided. Your exam will **not be graded** without your signature/name listed below:

I certify that my responses in this examination are solely the product of my own work and that I have fully abided by the Bryn Mawr College Academic Integrity policy and instructions stated above while taking this exam.

Signed:

For administrative use only!

1	2.1	2.2	2.3	3.1	3.2	3.3	4.1	4.2	Total

**Part 1: Syntax (1 point each; 15 points total)**

In the space provided, write the Java instruction that would do each of the following. Your code must be syntactically correct; no partial credit will be awarded.

**(1.1)** Declare, but do not initialize, an `int` variable named `k`.

**(1.2)** Set the value of the variable `k` (from above) to 5.

**(1.3)** Declare a `double` variable named `m` and initialize it to 37.8; do this as a single instruction.

**(1.4)** Increment the value of the variable `m` (from above) by the value of the variable `k` (from above).

**(1.5)** Convert the value of `m` to an `int` and then store its value in `k`; do this as a single instruction.

**(1.6)** Set the value of the variable `m` to the square root of the variable `k`.

**(1.7)** Set the value of the variable `k` to be a random integer in  $[0, 5]$ , i.e. between 0 and 5, inclusive.

**(1.8)** Assume you have `int` variables `a`, `b`, and `c`. Set the value of the variable `k` to be equal to `a` times the sum of `b` and `c`; do this as a single instruction.

**(1.9)** Assume you have **boolean** variables **x** and **y**. Declare a **boolean** variable **n** and initialize it so that its value is true if both **x** and **y** are true, but is false otherwise; do this as a single instruction.

**(1.10)** Assume you have an **int** variable **p**. Set the value of **n** (from above) to be true if **p** is even and false if **p** is odd; do this as a single instruction.

**(1.12)** Declare and initialize a variable called **nums** as an array of 10 **ints**.

**(1.13)** Set the element at index #5 of the **nums** array (from above) to 113.

**(1.14)** Set the last element of the **nums** array to be equal to 8.

**(1.15)** Assume you have an array named **vals**. Set the value of the variable **k** (from above) to the number of elements in the **vals** array.

**Part 2: Analysis & Debugging (10 points each; 30 points total)**

**(2.1)** What are the values of `x` and `y` after the following code is run (i.e., after line 18 is reached)? Note that the line numbers on the left are *not* part of the program.

```
1  int x = 0, y = 1, z = 0;
2
3  if (x >= y) {
4      z = 1;
5  }
6  else if (z == 1) {
7      x = y;
8  }
9
10 if (y == 1) {
11     if (x == 0 || z == 0) {
12         y = x + z;
13     }
14     x = 2;
15 }
16 else {
17     x = 5;
18 }
```

x is:

y is:

**(2.2)** How many times is "Hello!" printed in the following code? Note that the line numbers on the left are *not* part of the program.

```
1  int n = 5;
2
3  for (int i = 0; i < n; i++) {
4      for (int j = i; j < n; j++) {
5          System.out.println("Hello!");
6      }
7  }
8
9
10 }
```

Enter your answer here:

**(2.3)** Consider the following algorithm: assuming that `ints a` and `b` have been declared and initialized:

1. declare an `int` called `c` and initialize it to 0
2. as long as `a` is greater than or equal to `b`, do the following:
  1. increment `c` by 1
  2. set `a` to be equal to `a` minus `b`

Implement the above algorithm in Java. You do not have to write an entire program, just the part described above.

After executing this code, what is the value of `c` in terms of `a` and `b`? You can describe it in English or as a Java expression.

**Part 3: Modifying Code (10 points each; 30 points total)**

**(3.1)** Assume that **A** and **B** are `int` arrays that have been declared and initialized, and that each element in **A** is distinct, and that each element in **B** is distinct. The following code is attempting to count the number of elements that **A** and **B** have in common; note that the line numbers on the left are *not* part of the program.

```
1  int count;
2
3  for (int a : A) {
4
5      for (int b : B) {
6
7          if (a = b) {
8
9              count += 1;
10
11          }
12      }
13 }
```

There are exactly **two** errors in this code that would cause it not to compile. In the space below, identify the line numbers on which the errors occur, and how you would fix them so that the code works as expected. Be sure to specify the exact code you would write in order to fix the problems.

Line number	How would you fix it?

**(3.2)** In the space below, rewrite the following code so that it maintains the same behavior but uses a **while**-loop instead of a **for**-loop. You can assume that the variables **n**, **k**, and **vals** have all been declared and initialized:

```
for (int i = 0; i <= n; i++) {  
    vals[i] = Math.pow(k, i);  
}
```

**(3.3)** In the space below, rewrite the following code so that it maintains the same behavior but uses **if-else** statements instead of a **switch** statement. You can assume that the variable **x** has been declared as an **int** and has been initialized.

```
int a;  
switch (x) {  
    case 0:  
        a = 1;  
        break;  
    case 1:  
        a = 0;  
        break;  
    default:  
        a = -1;  
}
```

**Part 4: Writing Code (25 points total)**

**(4.1; 10 points)** Assume that `nums` is an `int` array that has been declared and initialized. Write code that would display the elements of `nums` in reverse order using `System.out.println()`. That is, it should first display the last element, then the second-to-last, and so on. You do not need to write an entire program, just the code that performs this operation.



If you need additional space for any of your responses, you may include it here. Be sure to make it clear which question you're answering!

A large, empty rectangular box with a black border, intended for providing additional space for responses to questions. The box is currently blank.