



**1. Matching [ 10 pts ]**

Choose the **best** word for each statement.

1. \_\_\_\_\_ The file type of Java source files.
2. \_\_\_\_\_ Converts a high level programming language into one that the computer can interpret.
3. \_\_\_\_\_ The name of the java virtual machine.
4. \_\_\_\_\_ Has no return type.
5. \_\_\_\_\_ Keyword used to declare constants in Java.
  - A. Object Construction
  - B. final
  - C. Byte Code
  - D. Space Tracker
  - E. java
  - F. javac
  - G. Public method
  - H. Encapsulation
  - I. Private method
  - J. class
  - K. Photon
  - L. Feedback Control

**2. Java Data Types [ 10 pts ]**

Circle all of the words that **are not** Java primitive types.

for                      String                      int                      triple                      do

polymorphism                      Object                      double                      short                      byte

long                      wide                      boolean                      java                      char

### 3. Declaration and Initialization [ 10 pts ]

Indicate whether or not the following statements are syntactically correct, ie they will compile correctly. If the statement is correct, write **OK** and proceed to the next part; if the statement is incorrect, write **ERROR** and re-write the statement so the problem is fixed. If you decide to write OK and make a fix to a part, that part will be marked entirely wrong.

2 (a) `int retVal = 4;`

2 (b) `String int = "Wicked Awesome";`

2 (c) `double d;`

2 (d) `float a=23.0f;`

2 (e) `String temp= 90;`

#### 4. **Compilation and Execution [10 pts]**

- 5 (a) What command do you type at the command prompt to compile all of the java files in the directory?
- 5 (b) What command do you type at the command prompt to run the java class **GoClass**?

## 5. Loss of Precision and Casting [ 10 pts ]

Given the following statements determine where a cast is needed. If no casting is needed, write **OK**; if casting is required because of possible loss of precision write **ERROR** and change the code so it compiles. No extraneous casts should be made: that means if it will compile without casting, then you shouldn't add a cast just in case.

- 2 (a) `double b = 9.19;`  
`short a = (short) b;`  
`int c = b;`
- 2 (b) `int trouble = 45;`  
`char gobble = trouble;`
- 2 (c) `double myDouble = 32;`
- 2 (d) `double partd = 32.0f;`  
`float partd2 = partd;`
- 2 (e) `double lastone = 32.0f;`

6. **Arrays [ 10 pts ]**

Write the method `public double findMax(double[] b)` which returns the largest value inside the array. You can assume that the array is of non-zero length.

```
public double findMax(double[] b) {
```

```
}
```

## 7. Iteration [ 10 pts ]

Write the method **public int odds()** which returns the count of **odd** numbers between 213 and 509595. 213 and 509595 should also be considered.. You **must use iteration** to solve this problem. If you do not use iteration, or if you use any recursion, you will receive no credit.

```
public int odds() {
```

```
}
```

**8. Accessors and Mutators [ 10 pts ]**

Given the following class framework, write accessors and mutators for all the variables:

```
public class p8{  
  
    private String peace;  
    private byte thought;  
    private int force;
```

```
} //end class p8
```

**9. Loops [ 10 pts ]**

Convert the following **while loop** into a **for loop**.

```
int j=22;
while(i >=4){
    i-=2;
    System.out.println(i);
    System.out.println(--i);;
}
```

10. **Tracing [ 10 pts ]**

Given the following class, write the output when the class is run.

```
public class TraceB{
    public String printMe(){
        return "Wow";
    }//end printMe()

    public static void main(String[] argv){
        boolean a = true;
        boolean b = false;
        TraceB c = new TraceB();
        int j = 7;
        if( a && b ){
            System.out.println("Solo");
            a=false;
        }
        if( a || b ){
            System.out.println("U2");
        }
        do{
            j--;
            System.out.println("Look out");
        }while(j>=5);
        System.out.println(j);
        c.printMe();
    }//end main(String[])
}//end class TraceB
```

---