READ THIS NOW!

Failure to read and follow the instructions below may result in severe penalties.
Failure to adhere to these directions will not constitute an excuse or defense.

- Print your name in the space provided below.
- Print your name and ID number on the Opscan form; be sure to code your ID number on the Opscan form. Code **Form A** on the Opscan.
- Choose the single best answer for each question — some answers may be partially correct. If you mark more than one answer, it will be counted wrong.
- Unless a question involves determining whether given Java code is syntactically correct, assume that it is. The given code has been compiled and tested, except where there are deliberate errors. Unless a question specifically deals with Java import directives, you should assume the necessary library files have been imported.
- Note that questions about printed values disregard formatting completely.
- In questions/answers which require a distinction between integer and real values, integers will be represented without a decimal point, whereas real values will have a decimal point, [1054 (integer), 1054.0 (real)].
- When you have completed the test, sign the pledge at the bottom and turn in the test and your Opscan.
- **This is a closed-book, closed-notes examination. No calculators or other electronic devices may be used during this examination. You may not discuss (in any form: written, verbal or electronic) the content of this examination with any student who has not taken it. You must return this test form when you complete the examination. Failure to adhere to any of these restrictions is an Honor Code violation.**
- There are 25 multiple-choice questions.
- Mark your answers on the test form, for future reference, and on the Opcan. The answers you mark on the Opscan form will be considered your official answers.

Do not start the test until instructed to do so!

Name ______________________________ **(print: Last name, First)**

VT PID: ______________________________ **(print: campus email address)**

Pledge: On my honor, I have neither given nor received unauthorized aid on this examination.

KEY
____
signature

1) According to Polya's 4-step process, what is the first thing to do?

1) Devise a plan　　2) Write Java code　　**3) Understand the problem**

4) Test the plan　　5) call 4-HELP　　6) none of these

---

2) In the Java language, how many classes can be instantiated from an object declaration?

1) 1　　2) 2　　3) 3
5) 5　　6) 6　　7) 7
9) 9　　**10) 0**

Objects are instantiated from classes, not vice-versa.

---

3) In the Java language, the behaviors of an object represent what ?

1) the attributes of the object　　2) the tasks the object can perform

3) the instance variables of the object　　4) the methods of the objects

5) both 1 & 3　　**6) both 2 & 4**　　7) none of the above

---

4) Which one of the following statements describing objects and classes is **true**?

1) Objects are instances of classes.

2) Classes are instances of Objects.

3) Objects and Classes are instances of structures.

4) Reference variables refer to classes.

5) All of the above are true.

---

5) Communication between objects is accomplished by which of the following object-oriented language features?

1) attributes　　2) messages　　3) instance variables

4) Java Virtual Machine　　5) none of the above

---

6) In which one of the following files must the Java method named `main` be stored?

1) main.class　　2) main.j　　3) main.java

4) main.c　　**5) none of the above**

The main method is not required to be in a particular class, or a specific file.

7) Which one of the following files would the Java compiler create from correct translation of the Java class named `Test1` ?

1) `Test1.j` **2) `Test1.class`**

4) `Test1.obj` 5) `none of the above`

Java bytecode produced by a Java compiler is stored in a file with the same name as the class and the extension of the file is "`.class`" . Some compilers will automatically bundle the <bytecode>.class file in a "`.jar`", which is a Java compressed file format.

8) What is the value printed for the variable `alpha` if the following code is executed?

```
String alpha;
alpha = "test1";
String beta = alpha;
alpha = beta.replace('t', 'b');
System.out.println(alpha);
```

1) `best1` **2) `besb1`**

4) `btesbt1` 5) `tbestb1`

String objects are immutable, they cannot be changed. New string objects can be instantiated and returned by string member functions, but they do not modify existing string objects. The reference variable alpha is re-assigned to refer to the string object returned by the String member function replace() which returns a copy of the invoking String object with all occurrences of the first character parameter in the string replaced by the second character parameter.

9) In the following Java statement:

```
System.out.print("test1".toUpperCase());
```

Which of the following is the stream reference in the above statement?

1) `System.out` 2) `PrintStream` 3) `print`

4) `test1` 5) `toLower` 6) None of these

System.out is the reference to the standard output stream , (the monitor).

10) What is the value printed for the variable `beta` if the following code is executed?

```
String alpha, beta;
alpha = "test1";
beta  = alpha;
alpha = beta.toUpperCase();
System.out.println(beta);
```

1) `test1` 2) `TEST1`

4) `alpha` 5) `ALPHA`

String objects are immutable. The reference variable alpha is assigned to refer to the string literal object "test1'. beta is assigned to alpha which makes beta an alias reference to the string literal object "test1'. The String function toUpperCase() returns a new String object to which alpha is set to reference. However, this does not affect the string literal object "test1' to which beta refers at the time of output.

11) How many unique object references are there in the following segment of code?

```
String gamma;
gamma = "test1".toUpperCase();
System.out.println(gamma);
```

1) `1` 2) `2` 3) `3`

4) `4` 5) `5` 6) 0

12) What is the value printed for the variable `epsilon` if the following code is executed?

```
String alpha, epsilon;
alpha = new String("Gobblers");
epsilon = alpha.substring(3,7);
System.out.println(epsilon);
```

Recall that the first character index position of strings in Java is zero. This version of the substring() function returns a reference to a new string object which contains all of the characters starting at index 3 of the parameter string and up to but not including the character at index 7 of the string.

1) `bbler`　　2) `bble`

4) `bler`　　5) `bblers`　　6) None of these

---

13) What is the value printed for the variable `delta` if the following code is executed?

```
String alpha, delta;
alpha = new String("Gobblers");
delta = alpha.substring(3);
System.out.println(delta);
```

This version of the substring() function returns a reference to a new string object which contains all of the characters starting at index 3 of the parameter string.

1) `Gob`　　2) `bbl`　　3) `bblers`

4) `ble`　　**5) `blers`**　　6) None of these

---

14) What is the value printed for the variable `sigma` if the following code is executed? (Note: The quotes in the responses below are not output. They are used to delimit the output.)

```
String sigma;
sigma = "  ttHOKIEStt  ";
sigma = new String("  tttVTttt  ");
System.out.println(sigma.trim());
```

The substring() function trim() returns a reference to a new string object which is a copy of the invoking String object but all of the leading and trailing whitespace characters are deleted. This would result in the returning of the "`tttVTttt`" string.

1) " `HOKIES` "　　2) "`HOKIES`"

4) " `VT` "　　5) "`VT`"　　**6) None of these**

---

15) Recall that in interactive input/output in Java a prompt is a string of text that is displayed/printed to let a user know what to type/enter at the keyboard. For example,

```
        System.out.print("Enter your Name: ");
```

which of the following lines of code should be executed after the above prompt output statement to guarantee that the prompt text will always be displayed before the user must enter the data?

1) `System.out.prompt();`　　2) `System.out.display();`

3) `System.out.flush();`　　4) `System.out.new();`

5) `System.out.println();`　　6) None of these.

---

16) How many constructor methods can a Java class contain?

1) `1`　　2) `2`　　3) `3`

4) `0`　　**5) None of these.**

A class may contain any number of constructor methods.

For the next two questions use the responses below:

1)

```
BufferedReader test1Input;
test1Input = new BufferedReader(
                new InputStreamReader(
                   new File("exam1.txt")));
String inLine; //input string
inLine = test1Input.readLine();
```

2)

```
PrintStream fileOut;
fileOut = new PrintStream(
              new StreamOutputFile(
                   new File("test1.txt")));
fileOut.println("Test 1");
```

3)

```
BufferedReader test1Input;
test1Input = new BufferedReader(
                new FileInputStream(
                   new File("exam1.txt")));
String inLine; //input string
inLine = test1Input.readLine();
```

4)

```
PrintStream fileOut;
fileOut = new PrintStream(
              new FileOutputStream(
                   new File("test1.txt")));
fileOut.println("Test 1");
```

5) None of the above

See the course notes and textbook if you missed this question.

17) Which of the above code segments would print a line of text into the disk file `"test1.txt"`? **4**

18) Which of the above code segments would read a line of input from the disk file `"exam1.txt"`? **5**

Given the following Java reference declarations:

```
          String VT = new String(“ Va Tech Hokies ”);
          String hokie;
```

For the next two questions use the responses below:

1)

```
hokie = VT.trim().toUpperCase();
```

2)

```
hokie = new String(“ Va ”).concat(new String(“Tech ”)
                          .concat(new String(“Hokies ”)));
```

3)

```
hokie = new String(“ Va ” + “Tech ”  + “Hokies ”);
```

4)

```
hokie = “ Va ” + “Tech ”  + “Hokies ”;
```

5)

None of the above

19) Which of the above code segments is an example of object creation through the process of cascading? 4

20) Which of the above code segments is an example of object creation through the process of composition? 5

Composition is the process of sending a message to an object to create a new object whose reference is used as an argument in a message. In turn, this may yield a reference to another new object, which then could be used as an argument in yet another message, and so on. The results of messages are used as arguments in additional messages.

Cascading is the process of sending a message to an object to create a new object, which in turn is sent to a message to create another new object, which in turn is sent to a message to create yet another new object, and so on.

For the next three questions, assume the following class declaration and implementation:

```
class Flashlight {

public  Flashlight (){
   light   = “off”; //light off
   battery = 5;     //full charge
}

public Flashlight (String state,
                   int charge);
   light   = state;
   battery = charge;

public void on() {
   light = “on”; //turn on
}

public void off() {
   light = “off”; //turn off
}
```

```
public String onoff() {
   return light; //return state
}

public void recharge(int charge) {
   battery = battery + charge;
}

public int power() {
   return battery; //return charge
}

private String  light;//”on”==light on
private int  battery; //0 depleted

}
```

Instantiating an object without passing parameters to the constructor results in the default (parameterless) constructor being invoked. Which in this case gives the state of the constructed object in the comments.

21) What does the following statement accomplish:

```
Flashlight Keyring = new FlashLight();
```

1) define an instance of the class `Keyring`

2) define an instance named `Keyring` of a class `Flashlight` with unknown status

3) define an instance named `Flashlight` of a class `Keyring` with unknown status

4) define an instance named `Keyring` of a class `Flashlight` with its light off and battery fully charged

5) define an instance named `Flashlight` of a class `Keyring` with its light off and battery fully charged

6) None of these

22) What does the following statements accomplish:

```
Flashlight Compact;
Compact = new Flashlight(“on”, 1);
```

1) define an instance of the class `Compact`

2) define an instance named `Compact` of a class `Flashlight` with unknown status

3) define an instance named `Flashlight` of a class `Compact` with unknown status

4) define an instance named `Compact` of a class `Flashlight` with its light on and battery minimally charged

5) define an instance named `Flashlight` of a class `Compact` with its light on and battery minimally charged

6) None of these

Instantiating an object by passing parameters to the constructor results in a signature matching (parameterlized) constructor being invoked. Which in this case turns the flashlight on and sets its to one which is one above the depleted state.

23) What do the following statements accomplish:

```
Flashlight Belt;
Belt = new Flashlight(“off”, 5);
Belt.recharge(-5);
```

1) requests the `Flashlight` object `Belt` to fully charge its battery
2) **requests the `Flashlight` object `Belt` to completely discharge its battery**
3) requests the `Flashlight` object `Belt` to turn on and stay on until its battery is completely discharged
4) the statement contains a syntax error
5) None of these

The recharge method adds its parameter to the battery value, which in this case reduces it to zero, resulting in a depleted battery.

---

24) What is the lifetime of an object?

1) objects exist from the point of their creation until the end o
2) objects exist from the point of the declaration of their reference variable and until their reference variable goes out of scope.
3) objects exist from the point of the declaration of their reference variable and until the containing Java program completes execution
4) **objects exist from the point of their creation until it is no longer referenced.**
5) None of these

Objects are garbage collected by the Java virtual machine when no variables reference them.

---

25) What is the lifetime of local variables?

1) **local variables exist from the point of their declaration until the end of the block in which they are declared.**
2) local variables exist from the point of the declaration and until their containing object goes out of scope.
3) local variables exist from the point of the declaration and until the containing Java program completes execution
4) local variables exist from the point of their declaration and until they are no longer referenced.
5) None of these

Local variables, such as those in function blocks, exist until they go out of scope, (access range), which occurs when the function goes out of scope which happens when the function execution ends.