1. _________________

2. type _________________
   value _________________

3. type _________________
   value _________________

4. (a) _________________
   (b) _________________
   (c) _________________

5. a b c d
6. a b c d
7. a b c d

8. return statement ________
   actual parameters ________
   return type ________
   formal parameters ________

9. (a) _________________
   (b) _________________
   (c) _________________
   (d) _________________

10. (a) _________________
    (b) _________________

11. (a) _________________
    (b) _________________
    (c) _________________
    (d) _________________

12. _________________
This is a closed-book, no-calculator, no-electronic-devices, individual-effort exam. You may reference one page of handwritten notes. All answers should be clearly written. Questions that require code should be written using correct Java syntax. You may write \texttt{SOP} to represent \texttt{System.out.println}.

<table>
<thead>
<tr>
<th>Class</th>
<th>Method/Constructor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scanner</td>
<td>Scanner(System.in)</td>
<td>create Scanner for parsing System.in</td>
</tr>
<tr>
<td></td>
<td>Scanner(String text)</td>
<td>create Scanner for parsing text</td>
</tr>
<tr>
<td></td>
<td>String next()</td>
<td>get next delimited word</td>
</tr>
<tr>
<td></td>
<td>double nextDouble()</td>
<td>get next delimited double</td>
</tr>
<tr>
<td></td>
<td>boolean nextBoolean()</td>
<td>get next delimited boolean</td>
</tr>
<tr>
<td></td>
<td>int nextInt()</td>
<td>get next delimited integer</td>
</tr>
<tr>
<td>String</td>
<td>int length()</td>
<td>get number of characters</td>
</tr>
<tr>
<td></td>
<td>char charAt(int i)</td>
<td>get the character at index i</td>
</tr>
<tr>
<td></td>
<td>String toUpperCase()</td>
<td>get a String like this one, but in all-caps</td>
</tr>
<tr>
<td></td>
<td>int indexOf(char c)</td>
<td>find the index of the first c</td>
</tr>
<tr>
<td></td>
<td>String substring(int a)</td>
<td>get substring from index a to String's end</td>
</tr>
<tr>
<td></td>
<td>String substring(int a, int b)</td>
<td>get substring from index a to before index b</td>
</tr>
<tr>
<td>Math</td>
<td>int max(int a, int b)</td>
<td>get the maximum of a and b</td>
</tr>
<tr>
<td></td>
<td>int min(int a, int b)</td>
<td>get the minimum of a and b</td>
</tr>
<tr>
<td></td>
<td>double pow(double base, double exponent)</td>
<td>raise base to the exponent power</td>
</tr>
<tr>
<td>Random</td>
<td>Random()</td>
<td>create a random number generator</td>
</tr>
<tr>
<td></td>
<td>nextInt(int i)</td>
<td>get random number in $[0, i - 1]$</td>
</tr>
<tr>
<td></td>
<td>nextDouble()</td>
<td>get random number in $[0.0, 1.0)$</td>
</tr>
<tr>
<td>Color</td>
<td>Color(int r, int g, int b)</td>
<td>create a color, with each intensity in $[0, 255]$</td>
</tr>
</tbody>
</table>
1. What is the type of the following expression?

`'g'`

2. What are the type and value of the following expression?

`100 % 3`

3. What are the type and value of the following expression?

`"hundred".length() / 2.0`

4. List three object types that we have met in lab, lecture, or homework.

5. Which of the following code sequences use *unnecessary* casts? Check zero or more.

   (a) `int average = (int) (('a' + 'z') / 2);`

   (b) `int i = 32767;
       short s = (short) i;`

   (c) `Scanner in = new Scanner(System.in);
       long a = in.nextInt();
       int b = (int) a;`

   (d) `double x = (double) new Random().nextInt();`

6. You see the name `DataOutputStream` in code that follows Java coding conventions. What could `DataOutputStream` be? Check zero or more.

   (a) a variable

   (b) a class

   (c) a constant

   (d) a method name

7. You see the name `tally` in code that follows Java coding conventions. What could `tally` be? Check zero or more.

   (a) a method name

   (b) a constant

   (c) a class

   (d) a variable

8. Identify the anatomical parts of a method by writing the letter of the blank next to the part’s name.

   ```java
   public static void main(String[] args) {
       jump(____a____);
   }

   public static ____b____ jump(_____c_____) {
       // ...
       _____d____
   }
   ```
9. Identity each of the following “chunks” of Java code as one of expression, statement, or declaration. Use the most descriptive term.

(a) 7 * 9 % 10
(b) System.out.println("To begin, begin.");
(c) int mp;
(d) in.nextInt()

10. Fill in blanks a and b with the best answer given the surrounding code.

```
public static ____a____ around(____b____) {
    return String.format("%d, %d", n - 5, n + 5);
}
```

11. Fill in blanks a, b, c, and d with the best answer given the surrounding code.

```
public static ____a____ charAtBackward(____b____, int i) {
    ____c____ = s.length();
    ____d____ = s.charAt(len + i);
    return c;
}
```

12. Put the following scrambled code back in order:

```
a: Color rgb = new Color(in.nextInt(), in.nextInt(), in.nextInt());
b: public static Color swap(String triplet) {
    c: in.useDelimiter(\",\", \",\",);
    d: }
    e: Scanner in = new Scanner(triplet);
    f: return rgb;
```

The letters to the left are just labels, not actual Java code. For your answer, write the letter of the first line of code, then the letter of the second, and so on. For example: cfabde.

13. Write a main method that prompts and retrieves from the user a month name and a four-digit year. Print the first three letters of the month name (which you can assume has at least three letters) and the last two digits of the year. Model your program on this example console interaction:

```
Enter month: October
Enter year: 2017
Oct17
```


15. Write a method named insert. It accepts as parameters two Strings and an int index. It returns a new String like the first String, but with the second String inserted at the given index. For example, insert("meaning", "der", 4 -> "meandering". Assume the index is valid.