```
A.   ====================================================
B.   public abstract class Text
C.   {
D.     protected String author;
E.     public Text(String a)  { author = a; }
F.     public String getAuthor()   { return author; }
G.     public void narrate()  { System.out.println("");}
H.   }
I.   ====================================================
J.   public class Poem extends Text
K.   {
L.      protected String title;
M.      public Poem (String a, String t)  {
N.        super(a);
O.        this.title = t;
P.      }
Q.
R.      public void narrate () {
S.       System.out.println("The title is ” + title);
T.       System.out.println(“and author is "+ getAuthor());
U.      }
V.   }
W.   ====================================================
X.   public class Haiku extends Poem
Y.   {
Z.     public void recite()    {
AA.      narrate();
BB.      System.out.println("I am 17 syllables long.");
CC.    }
DD.  }
EE.  ====================================================
FF.  final class DRIVER {
GG.     public static void main(String [] args)   {
HH.       Haiku h = new Haiku (“Basho”,”Spring Finals”);
II.       Poem p = new Poem (“Frost”,”Fire and Ice”);
JJ.       h.recite();
KK.       p.narrate ();
LL.       h=p;
MM.       System.out.println(h.getAuthor());
NN.       System.out.println(p.getAuthor());
OO.  }    }
```

Name_____

1. Explain what the code in Line B above indicates:
   Class definition.
   Abstract classes cannot be instantiated.

2. Explain what the code in Line **D** above indicates:
   Field author, type string.
   Protected: private to world but public to descendants

3. Explain what the code in Line E above indicates:
   Public Constructort takes a String parameter

4. Explain what the code in Line F above indicates:
   Public method getAuthor, no params, returns a string
   An accessor

5. Explain what the code in Line G above indicates:
   Public method narrate, no params, returns nothing

6. Explain what the code in Line J above indicates:
   Class Poem inherits all Text functionality and fields

7. Explain what the code in Line L above indicates:
   Field author, type string.
   Protected: private to world but public to descendants

8. Explain what the code in Line M above indicates:
   Constructor that takes two String params, public

9. Explain what the code in Line N above indicates:
   Call to parent constructor, Text (Line E)

10. Explain what the code in Line O above indicates:
    this.title is the field on line L

11. Explain what the code in Line R above indicates:
An override of the narrate function of the parent class

12. Explain what the code in Line X above indicates:
Class Haiku inherits all of Poem
This means it also gets Text

13. Explain what the code in Line Z above indicates:
Public method recite, no params, or return

14. Explain what the code in Line AA above indicates:
Call to Poem. narrate();

15. Explain what the code in Line GG above indicates:
Main funtion s static, always exists .
Startting point for package execution

16. Explain what the code in Line HH above indicates:
Cretae new object 'h' , type Haiku.
Params Baso, Spring Finals, passed to constructor (Line M)

17. Explain what the code in Line II. above indicates:
Cretae new object 'p' , type Poem.
Params Frost, Fire and Ice, passed to constructor (Line M)

18. Explain what the code in Line LL above indicates:
'h' refernces object referenced by 'p'

19. What is the output of the program produced by Line JJ above:
```
The title is Spring Finals
and author is Basho
I am 17 syllables long
```

20. What is the output of the program produced by Line KK above:
```
The title is Fire and Ice
and author is Frost
```

21. What is the output of the program produced by Line MM above:
Frost

22. What is the output of the program produced by Line NN above:
Frost

23. Give a code fragment to test if **S** is the string **"gnu".**

```
if(s.equals("gnu")) z=1;
```

24. Write code that will set character **ch** to **'!'** if **x** equals **55** and will set **ch** to **'?'** otherwise?

```
if(x==55) s.o.p('!'); else s.o.p('?');
```

25. Write code that will set character **ch** to **'!'** if **x** is between **55** and **85**?

```
if(x>55 && x<85) s.o.p('!');
```

26. Write code that will set character **ch** to **'#'** if **X** is NOT between **55** and **85**?

```
if(!(x>55 && x<85)) s.o.p('!');
```

27. What is so special about an **Object** reference?

Can refer to ANY object.

28. What is the difference between 7, '7' and "7"?

int, char, string

29. What is the difference between **Integer** and **int**?

Class (wrapper class) and primitive data type

30. Write code to convert **string z="55";** to an integer and store the resulting value in a variable.

int iz= Integer.parseInt(z);

31. Write a for loop that prints out your name seven times.

for (i = 0; i <= 6; i++) System.out.println(“your name”);

32. How many times does the following loop execute?

```
for (i = 0; i <= 21; i++)
            System.out.println( i * i);
```

Answer:_____22_____

33. How many times does the following loop execute?

```
for (i = 14; i <= 21; i++)
            System.out.println( i * i);
```

Answer:_____8_____

34. How many times does the following loop execute?

```
for (i = 14; i <= 21; i--)
            System.out.println( i * i);
```

Answer:_____infinite?_____

35. What is a sentinel value?
Answer:_____Stops a loop when a special value is recognized_____

36. If we run the line of code **`System.out.println(12/5);`** what will the computer print out?
Answer:_____2_____

37. If we run the line of code System**`.out.println(12%5);`** what will the computer print out?
Answer:_____2_____

38. If we run the line of code **`System.out.println(12.0/5.0);`** what will the computer print out?
Answer:_____2.4_____

39. If we run the line of code **`System.out.println(12.0/5);`** what will the computer print out?2.
Answer:_____

40. If we run the line of code System.out.println(12/5.0); what will the computer print out?
Answer:_____2.4_____

41. If we run the line of code **`System.out.println( (int)12.0/5.0);`** what will the computer print out?
Answer:_____2.4_____

42. If we run the line of code **`System.out.println("2" + 5);`** what will the computer print out?
Answer:_____25_____

43. Write an expression to compute the value of $\frac{\sqrt{a^2+b^2}}{ab}$ and store the result in an appropriate variable called **`c.`**

double c = Math.sqrt((a*a+b*b)/(a*b));

44. What is a **`JComponent`**?

A thing on a GUI

45. What is a **`JFrame`**? `A rectangular area for building a GUI, Contains comonents and controls.`

46. What is an **`Applet`**?
`A GUI that runs in a web page.`

47. Assume **`g2`** is a graphics object.
Write a line of code to change the graphics object **`g2's`** color to red?
g2.setColor(Color.red);

48. Assume **`g2`**is a graphics object.
Write a line of code to change the graphics object **`g2's`** color to pink?
g2.setColor(Color.pink);

49. Assume **`g2`**is a graphics object.
What does **`g2.fill(new Rectangle(22,22,44,44,))`** do?
Paints interior of rectangle.

50. Assume **`g2`**is a graphics object.
What does **`g2.draw(new Rectangle(22,22,44,44,))`** do?
Paints border of rectangle.