CMPSC 111
Introduction to Computer Science I
Spring 2018
Janyl Jumadinova

Solutions to Final Exam Review Questions

1.
```
import java.util.Scanner;

public class Semi {
    public static void main(String[] args); {
        int i = 10,
            j = 20,
            k = 30;
        if (k > i+j); {
            System.out.println(k);
        };
        else {
            for (int m = 0; m < 10; m++); {
                System.out.println(m);
            }
        }
    }
}
```

2.

(a)
```
if (i = 3) // ==, not = {
  j = i+i;
}
```

(b)
```
if (i < 3); // No ";" {
  j = i+i;
}
```

(c)
```
if (i =< 1) // should be <={
  j = i+i;
}
```

(d)
```
if (i > 3) // "!(i>3)" or "i <= 3" {
  j = i+i;
}
```

(e)
```
if (i != 3) {
  j = i+i;
};; // no semicolon
else {
  j = i-i;
}
```

(f)
```
if ((i == 3) || (i == 4)) // add ()s {
  j = i+i;
}
```

(g)
```
if (i != 3 || i!= 4) // add text {
  j = i+i;
}
```

(h)
```
if (i ==2 && j == 2) // add text {
  i = i+i;
}
```

3.

```
for (int i == 3; i < 10; i++) // = {
  sum = sum + i;
}
```

(a)

```
int i = 10;
while (i < 3); // no semicolon {
  sum = sum + i;
}
```

(b)

```
for (int i <= 10 ) // wrong form {
  // for(int i=1;i<=10;i++)
  sum = sum+i;
}
```

(c)

```
for (int j=0; j<10; j=j+1) // ";"s {
  sum = sum + j;
}
```

(d)

```
int j = 3;
while (j = 3) // == or >= or... {
    sum = sum+j;
    if (sum % 7 != 0)
        j++;
}
```

(e)

```
for (int i = 0; i < 10; i++); // no ; {
  j = i+i;
}
```

(f)

```
int j = 10;
while ((j >= 10) && (j < 20)) // () {
  j = j+1;
}
```

(g)

```
for (int k = 0; k = 10; k++) // <= {
  System.out.print(k);
}
```

(h)

4.

`String s = "hello";`
`String t = 10; // "10"`
`String u = s + t;`
(a)
	`boolean b;`
	`int i = 20;`
	`b = false;`
	`int j = i + b; // can't add int and boolean`
	(b)

`int i = 10, j = 20;`
`i + 10 = j; // variable only`
(c)
	`double d = 5;`
	`int k = d; // can't use double`
	(d)

`int i = 10, j = 10.5, k = 11;`
`// can't use double`
(e)
	`Scanner scan = new Random();`
	`// wrong class`
	(f)

`char c = "A"; // 'A'` (g)	`Random rand = new Random( ); // ( )` (h)

`System.out.println("\");`
`// escape character`
(i)
	`int single = 1, double = 2, triple = 3;`
	`// reserved word`
	(j)

5.

`// "Get" method for x:`
`public int getX() {`
`  return x;`
`}`
	`// "Set" method for x:`
	`public void setX(int xNew) {`
	`  x = xNew;`
	`}`

`// "Get" method for y:`
`public double getY() {`
`  return y;`
`}`
	`// "Set" method for y:`
	`public void setY(double yNew) {`
	`  y = yNew;`
	`}`

`// "Get" method for z:`
`public String getZ() {`
`  return z;`
`}`
	`// "Set" method for z:`
	`public void setY(String zNew) {`
	`  z = zNew;`
	`}`

`// "Get" method for b:`
`public boolean getB() {`
`  return b;`
`}`
	`// "Set" method for b:`
	`public void setB(boolean bNew) {`
	`  b = bNew;`
	`}`

6. (a)

```
public Thing(int x1, double y1, String z1, boolean b1) {
    x = x1; // or this.x = x1;
    y = y1; // or this.y = y1;
    z = z1; // or this.z = z1;
```

```
            b = b1; // or this.b = b1;
        }
```

(b)
```
Thing t = new Thing(17, 5.5, "Hello", true);
```

7.
```
public static void main(String[] args) {
    Tree t1 = new Tree("oak",true);
    Tree t2 = new Tree("elm",true);
    Tree t3 = new Tree("pine",false);
}
```

8.
```
Alpha a = new Alpha(42,"large");
Beta b = new Beta(a);
System.out.println(b.getAlf());
```

9.

`count = 9` **(a)**	`count = 11` **(b)**
`count = 2+3+4+5 = 14` **(c)**	`count = 4` **(d)**
`count = 6` **(e)**	`count = 10` **(f)**
`count = 5` **(g)**	`count = 4` **(h)**

10.

`a = 30, b = 20` **(a)**	`a = 10, b = 30` **(b)**
`a = 5, b = 20` **(c)**	`a = 0, b = 40` **(d)**
`a = 20, b = 20` **(e)**	`a = 40, b = 30` **(f)**

11.
```
int sum = 0;
int odd = 1;
for (int count = 1; count <= 20; count++) {
    sum = sum + odd;
    odd = odd + 2;
}
```

12.
```
if ((x % 3 == 0) || (x >= 10 && x <= 20)) {
    System.out.println("yes");
}
else {
    System.out.println("no");
}
```

13.
```
int count = 0;
System.out.println("Enter a value (<= 0 to quit): ");
double value = scan.nextDouble();
while (value > 0) {
    if (value > 10.0) {
        count++;
```

```
        }
        System.out.print("Enter a value (<= 0 to quit): ");
        value = scan.nextDouble();
    }
```

14.

`finalexam` (of type `String`) **(a)**	`in` (of type `String`) **(b)**
`finalam` (of type `String`) **(c)**	`9` (of type `int`) **(d)**
`true` (of type `boolean`) **(e)**	`abcexam5` (of type `String`) **(f)**

15. (a) smallest: -5; largest: 4

    (b) Possible values are `'b'`, `'c'`, and `'d'`

    (c) `rand.nextInt(6)*2`

    (d) `rand.nextDouble()*20.0 - 10.0`

16.
```
int i = scan.nextInt();
String s = scan.next();
double d = scan.nextDouble();
```

17. (a) `sList.get(3)` equals `"frog"`

    (b) `sList.size()` equals 4

    (c) `sList.contains("at")` is `false`

    (d) `sList.get(0).contains("at")` is `true`

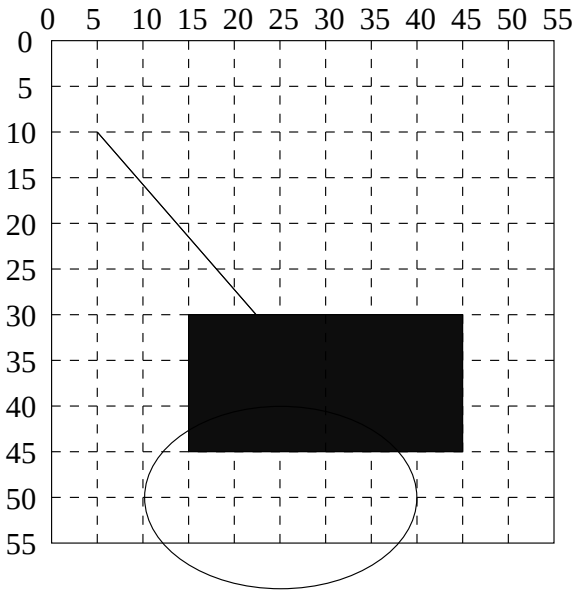
    (e) Final value is `m = "dogfrog"`

    (f)
```
for (int i = sList.size()-1; i >= 0; i--) {
    System.out.println(sList.get(i));
}
```

18.
```
ArrayList<Integer> iList = new ArrayList<Integer>();
for (int i = 0; i < 1000; i++) {
    iList.add(i);
}
```

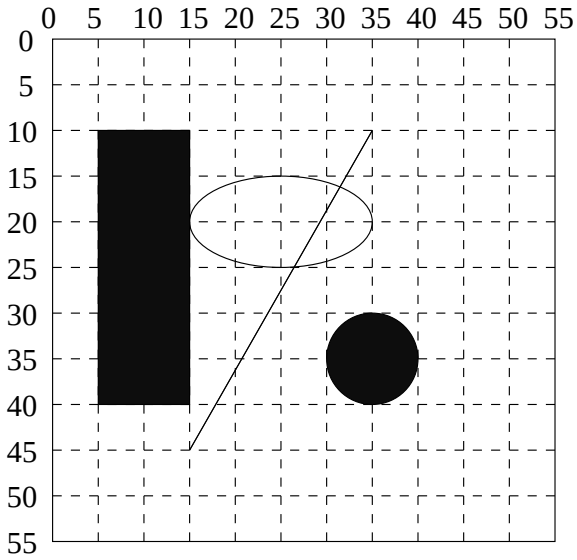
19.
```
for (int i = 0; i < 25; i++) {
    x[i] = 3*(i+1);
}
```
OR
```
for (int i = 1; i <= 25; i++) {
    x[i-1] = 3*i;
}
```

20. For part (a), draw the image corresponding to the Java code. For part (b), write the Java code corresponding to the image. In each part, assume that the `Graphics` object is named `page`. Just lightly scribble to show filled objects—don't worry about carefully shading in every little pixel!




```
page.fillRect(15,30,30,15);
page.drawLine(5,10,30,40);
page.drawOval(10,40,30,20);
```

(a)




```
page.fillRect(5,10,10,30);
page.drawOval(15,15,20,10);
page.drawLine(35,10,15,45);
page.fillOval(30,30,10,10);
```

(b)

21. Consider the three Java programs shown in the figure on the following page, together with the image produced by running `Draw.java`.

    (a) In method `paintComponent` of class `Forest`, add lines similar to ones already there:

    ```
    Tree t3 = new Tree(200,0);
    t3.draw(page);
    ```

    (b) Add an `import java.util.Random;` statement at the beginning of the `Forest` class. Add an instance variable near the top of the `Forest` class:

    ```
    private Random rand;
    ```

    Initialize it in the `Forest` constructor:

    ```
    rand = new Random();
    ```

    Replace the line "`Tree t1 = new Tree(0,0);`" with the following. Exact numbers are not important—anything between 200 and 300 is okay. (Note that 300 or more puts the tree off the screen.)

    ```
    Tree t1 = new Tree(rand.nextInt(200), rand.nextInt(200));
    ```

    (c) We could, for instance, create an instance variable in the `Forest` class as follows:

    ```
    private Tree tr[];
    ```

    We could initialize it in the `Forest` constructor as follows:

    ```
    tr = new Tree[5];
    ```

To add `Tree`s to the array, we would need to write things like:

```
tr[0] = new Tree(0,0);
tr[1] = new Tree(100,100);
tr[2] = new Tree(200,0);
   ... etc. for tr[3] and tr[4]
```

22. See the glossary at the end of your book for the definitions.

23. When we need to use a class such as `Random`, `Scanner`, `Date`, `ArrayList`, `Color`, `JFrame`, etc.—something that is not in a pre-loaded Java package but is one of the many, many libraries of special-purpose packages that are available in Java, we use an `import` statement to tell Java to include those classes or packages.