The exam will be closed-note and closed-book; please consider this fact before using your notes on this practice version. Please see the abbreviated Processing reference at the back of this exam.

This exam is intended to be taken in 80 minutes.

Assume that any given Processing statements do not have typographical errors. Aim for all code to be syntactically correct, but missing semi-colons (for example) will not be penalized.
Question 1 (10 points) Circle/write the correct answer in each of the following:

a) Which of the following is not one of the shape drawing commands in Processing:

1. quad(…)
2. triangle(…)
3. circle(…)
4. point(…)

b) What would Processing do as a result of this statement?
   line(width, 0, 0, height);

   Draw a line from the top-right corner to the bottom-left corner of the sketch

c) Consider the following function:

```java
void enigma(int k) {
    int answer = 0;
    if (k < 10) {
        if (k < 15) {
            answer = 1;
        }
        else {
            answer = 2;
        }
    }
    else if (k < 15) {
        answer = 3;
    }
    else {
        answer = 4;
    }
    println(answer);
}
```

For what argument values k does enigma(k) print the value 2?

1. enigma(k) will never print 2 for any value of k
2. k < 10
3. 10 <= k < 15
4. k >=15
d) How many times would this code print the text “Hello”?

```java
for (int i = 10; i >= 2; i-=2) {
    println(“Hello”);
}
```

5

e) What is the value of x after the following statements are evaluated?

```java
int y = 0, z = 1;
boolean x = (y >= 0 && z < 0);
```

false

f) Suppose that the variable x currently has the value 10. What statement would cause the value of x to become 13?

1. x + 3;
2. y = x + 3;
3. x = x + 3;
4. None. The value of x can't be changed.

g) What is the result of evaluating 9/2?

4

h) What is the result of evaluating 9%2?

1

i) What does the following code print?

```java
float x = 2/9;
println(x);
```

0.0

j) What does the following code print?

```java
float x = 9/2.0;
println(x);
```

4.5
Question 2 (5 points)
Write Processing statements to draw a red rectangle covering the right half of the sketch. Your statements should work for any sketch size.

```java
fill(255, 0, 0);
rect(width / 2, 0, width / 2, height);
```

Question 3 (5 points)
Write a conditional expression that would print “teenager” whenever an integer variable `age` has a value between 13 and 19, inclusive.

```java
int age = ...;
// write your conditional here:

if(age >= 13 && age <= 19)
{
    println("teenager");
}

// or the following:

if(age >= 13)
{
    if(age <= 19)
    {
        println("teenager");
    }
}
Question 4 (15 points)
The following Processing program is supposed to draw a blue circle of random diameter in the range 20 to 50 pixels centered on every mouse click. Complete the program by writing the appropriate commands needed to accomplish the task.

```java
void setup() {
    size(400, 400);
    background(255);
}

void draw() {
}

void mousePressed() {
    float size = random(20, 50);
    ellipse(mouseX, mouseY, size, size);
}
```
Question 5 (15 points)

a) Rewrite the following code, using a while-loop instead of a for-loop. The results of executing the two code fragments should be identical.

```java
int num = 6;
for(int i=10; i > 0; i-=2){
    println(num);
    num -= i;
}

int num = 6;
int i = 10;
while(i > 0)
{
    println(num);
    num -= i;
    i -= 2;
}
```

b) What is printed by the code fragment in 5(a) above?

- 6
- 4
- 12
- 18
- 22
Question 6 (15 points)
Write a Processing function that plays a simplified version of the Bizz/Buzz game. To do this, your code should print out consecutive numbers (starting at the parameter `start` and ending at `end`). Whenever a number is divisible by 5, it should print "Bizz" instead of the number. Whenever a number is divisible by 7, it should print "Buzz" instead of the number. Whenever a number is divisible by both 5 and 7, it should print "BizzBuzz" instead of the number.

(Note: The non-simplified version of the game also has special behavior when the digit 5 or 7 appears in the number. You do not have to worry about this part of the game.)

Remember that the % operator in Processing computes the modulus – that is, the remainder after dividing one integer by another.

```java
void bizzBuzz(int start, int end) {
    for(int i = start; i <= end; i++)
    {
        if(i % 5 == 0 && i % 7 == 0)
        {
            println("BizzBuzz");
        }
        else if(i % 5 == 0)
        {
            println("Bizz");
        }
        else if(i % 7 == 0)
        {
            println("Buzz");
        }
        else
        {
            println(i);
        }
    }
}
```
Question 7 (15 points)
Write a function named average that takes two integers x and y as parameters and use a loop to compute and print the average of all integers starting at x and ending at y, inclusive. For example, average(2, 5) should print 3.5, which equals (2+3+4+5)/4. You may assume that x < y.

```java
void average(int x, int y)
{
    float sum = 0;
    for(int i = x; i <= y; i++)
    {
        sum += i;
    }
    float avg = sum / (y - x + 1);
    println(avg);
}
```
Question 8 (20 points)
Consider the following code segment. Fill out the table that traces the function call to mystery.

```c
int a = 1;
int b = 2;
int c = 3;
int d = 4;

void setup() {
    d = 6;
    mystery(d);
}

void mystery(int e) {
    int f = 1;
    int result = 1;
    for (int i = 2; i < e; i++) {
        result = a + f;
        a = f;
        f = result;
    }
}
```

Trace of `mystery(d)`

<table>
<thead>
<tr>
<th>a</th>
<th>e</th>
<th>f</th>
<th>result</th>
<th>i</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>6</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>2</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>3</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>8</td>
<td>8</td>
<td></td>
</tr>
</tbody>
</table>

Directly after execution of the statement `mystery(d)`; in `setup()`, what are the values of the following variables?

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>2</td>
<td>3</td>
<td>6</td>
</tr>
</tbody>
</table>
**Extra Credit (20 points)** Write a complete Processing program to draw a clock face as shown below on the left. Partial credit for only drawing the hour (longer) ticks as shown on the right.

![Clock Face](image)

```java
void setup()
{
    size(500, 500);
    translate(250, 250);
    ellipse(0, 0, width, height);

    for(int i = 0; i < 60; i++)
    {
        line(240, 0, 250, 0);
        rotate(PI / 30);
    }

    stroke(255, 0, 0);
    for(int i = 0; i < 12; i++)
    {
        line(230, 0, 250, 0);
        rotate(PI / 6);
    }
}
```
PROCESSING QUICK REFERENCE

Variables/Constants:
width, height, mouseX, mouseY, HALF_PI, PI, QUARTER_PI, TWO_PI

Data Types
boolean, int, float

Conversion Functions
int(x), float(x), double(x), degrees(rad), radians(deg)

Relational and Logical Operators
!= (inequality) < (less than) <= (less than or equal to),
== (equality)  > (greater than)  >= (greater than or equal to)
! (logical NOT)  && (logical AND)  || (logical OR)

Loops
for (init; test; update) {
  statements
}
while (expression) {
  statements
}

Conditionals
if (test) {
  statements
}
if (expression) {
  statements
}
else {
  statements
}
else if (expression) {
  statements
}
...
else {
  statements
}

2D Primitives
arc(x, y, width, height, startRad, endRad)
ellipse(x, y, width, height)
line(x1, y1, x2, y2)
point(x, y)
quad(x1, y1, x2, y2, x3, y3, x4, y4)
rect(x, y, width, height)
triangle(x1, y1, x2, y2, x3, y3)
text(message, x, y)
PROCESSING QUICK REFERENCE

**Attributes**
ellipseMode(mode), rectMode(mode)

**Vertex**
beginShape(), endShape(), vertex(x,y), curveVertex(x,y)

**Mouse**
mouseClicked(), mouseDragged(), mousePressed(), 
mouseReleased(), mouseX, mouseY

**Text Output**
print(message), println(message)

**Setting**
background(r,g,b), fill(r,g,b), noFill(), noStroke(), 
stroke(r,g,b)

**Trigonometry**

\[ \cos(\text{rad}), \sin(\text{rad}), \tan(\text{rad}) \]

**Random**
random(lower,upper)