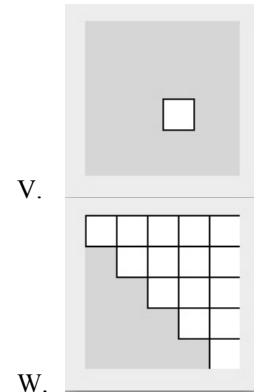
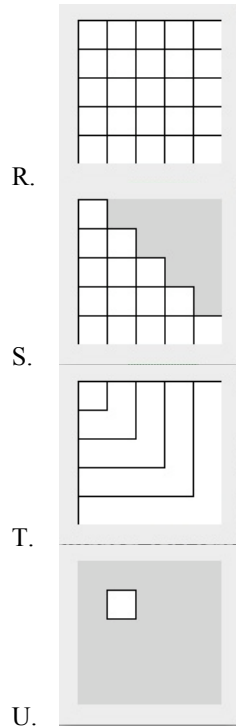
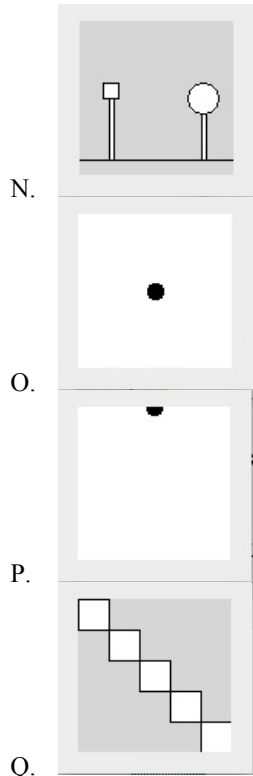


This is a closed notes, closed book exam. Each problem is worth 1 point unless indicated otherwise.

There are 21 problems and 25 points total. There are multiple versions of this exam. Don't be tempted to try and copy.

The first set of nine problems is an extended "matching" problem. Each problem shows a Processing sketch. Enter the letter that corresponds to the output. The possible choices are not all used and some *may* be used more than once.

- A. A black circle moves from right to left across the display.
- B. A black circle moves from left to right across the display.
- C. A black circle moves from top to bottom across the display.
- D. A black circle moves from bottom to top across the display.
- E. A black circle moves repeatedly back and forth (side-to-side) across the display.
- F. A black circle moves repeatedly up and down (top-to-bottom) across the display.
- G. A square area filled with small squares with the lower right corner of the filled area near $(\max(\text{mouseX}, \text{mouseY}), \max(\text{mouseX}, \text{mouseY}))$.
- H. A square area filled with small squares with the lower right corner of the filled area near $(\min(\text{mouseX}, \text{mouseY}), \min(\text{mouseX}, \text{mouseY}))$.
- I. A square area filled with small squares. The filled area starts with just one square then increases in size each frame until it fills an area determined by the current mouse location, then it starts over again with just one square.
- J. A black circle moves from the top center to the middle of the display and then moves diagonally to the lower right corner.
- K. A black circle moves from the bottom center to the middle of the display and then moves diagonally to the upper right corner.
- L. A black circle moves in a straight line from the top center to the lower right corner.
- M. A black circle moves in a straight line from the bottom center to the upper right corner.



1. Enter the best choice from the list of choices A-W above for what this program displays _____.

```

line(0,90,100,90);
rect(19,50,3,40);
rect(79,50,3,40);
rect(15,40,10,10);
ellipse(80,50,20,20);

```

2. Enter the best choice from the list of choices A-W above for what this program displays _____.

```

for (int row = 0; row < height; row+=20) {
  rect(row, row, 20, 20);
}

```

3. Enter the best choice from the list of choices A-W above for what this program displays _____.

```

for (int row = 0; row < height; row+=20) {
  for(int col = 0; col <= width; col+=20) {
    rect(col, row, 20, 20);
  }
}

```

4. Enter the best choice from the list of choices A-W above for what this program displays _____.

```

for (int col = 0; col < width; col+=20) {
  for(int row = 0; row <= col; row+=20) {
    rect(col, row, 20, 20);
  }
}

```

5. Enter the best choice from the list of choices A-W above for what this program displays _____.

```

for (int row = 0; row < height; row+=20) {
  rect(0, 0, width-row, height-row);
}

```

6. Enter the best choice from the list of choices A-W above for what this program displays _____.

```

// caution - this may be a trick question
void draw() {
  int ballPos = 0;
  background(255);
  int ballSize = 10;
  fill(0);
  ellipse(width/2, ballPos, ballSize, ballSize);
  ballPos = ballPos + 1;
}

```

7. Enter the best choice from the list of choices A-W above for what this program displays _____.

```

void setup() {
  frameRate(10);
}

int limit=0;
void draw() {
  background(255);

  if (limit > mouseX || limit > mouseY) limit = 0;

  for(int col = 0; col <=limit; col = col + 20) {
    for (int row = 0; row <= limit; row = row + 20) {
      rect(row, col, 20, 20);
    }
  }
  limit = limit + 20;
}

```

8. Enter the best choice from the list of choices A-W above for what this program displays ____.

```
// caution - this may be a trick question
int ballPos = 0;
int ballDirection = 1;

void draw() {
  background(255);
  int ballSize = 10;
  fill(0);
  ellipse(width/2, ballPos, ballSize, ballSize);
  ballPos = ballPos + ballDirection;
  if (ballPos < 0 || ballPos > 100) {
    ballDirection = ballDirection*(-1);
  }
}
```

9. Enter the best choice from the list of choices A-W above for what this program displays ____.

```
// caution - this may be a trick question
int xPos, yPos;

void setup() {
  yPos = 0;
  xPos = width/2;
}

void draw() {
  background(255);
  int ballSize = 10;
  fill(0);
  ellipse(xPos, yPos, ballSize, ballSize);

  yPos = yPos + 1;
  xPos = xPos + yPos/(height/2);
}
```

10. What does the following program print?

```
// caution - this may be a trick question
int[] data = new int[5];
for (int i = 0; i < data.length; i++) {
  data[i] = i*2;
}

for (int i = data.length-1; i >= 0; i--) {
  print(data[i] + " ");
}
```

11. What does the following program print?

```
int age = 92;
String ageBracket = "child";

if (age >= 65) {
  ageBracket = "senior";
}
if (age >= 21) {
  ageBracket = "adult";
}
if (age >= 16) {
  ageBracket = "youth";
}
println(ageBracket);
```

12. What does the following program print?

```
void setup {
  int x = 6, y = 3;

  printProduct(x, y);
  x = 2;
  print(" ");
  printProduct(x + y, x);
}

void printProduct(int x, int y) {
  print(x * y);
}
```

13. What does the following code fragment print?

```
for (int i = 1; i <= 40; i = i + 1) {
  if (i > 35)
    print(i + " ");
}
```

14. (1 point) How many times does the following program print "testing"? _____ (put answer here)

```
int i = 1;
if (i < 20 ) {
  while (i <= 30) {
    println("testing");
    i = i + 1;
  }
}
```

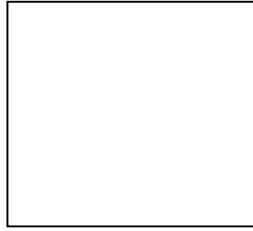
15. What is printed by the following program?.

```
void setup() {
  func2(2);
  func2(func3(3));
}
void func1(int x) {
  println("func1 " + x);
}
void func2(int x) {
  println("start");
  func1(x*3);
  println("end");
}
int func3(int x) {
  println("func3 " + x);
  return x*10;
}
```

16. What does the following program print?

```
void setup() {
  int x = 10;
  it = x;
  printIt();
  it = 20;
  printIt();
  printIt();
}

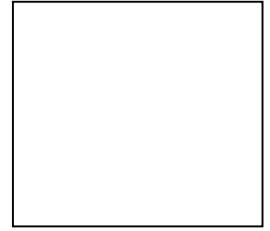
int it;
void printIt() {
  println(it);
  it = 0;
}
```



17. What does the following program print?

```
void setup() {
  int x = 10;
  printVal(x);
  printVal(20);
  println(x);
}

void printVal(int x) {
  println(x);
  x = 0;
}
```



18. (2 points) Assume that there is a global array of integers named `data`, write a code fragment that will count the number of zeros stored in the array and print out the count.

19. (2 points) Given the `Rocket` class from program 6 which includes instance variables `xVel` and `yVel`, write a method called `safeLanding()` that could be added to the `Rocket` class and returns `true` if the rocket made a safe landing and `false` otherwise. Assume the rocket has landed and you are only deciding if it was a safe landing or not. A safe landing is one in which both the `x` and `y` velocities are between `-0.4` and `0.4`.

```
class Rocket {
  float xVel, yVel;
  // most of the class is omitted and not needed for this question

  // put your safeLanding method here
}
```

20. (2 points) Fill in the blanks so this program creates two balls. Ball1 should start at location (50,100) with zero y velocity and 0.2 x velocity. Ball2 should start at (300,100) with zero y velocity and -0.2 x velocity. The balls should then fall according to gravity as specified in the Ball class. They should each be 10 pixels in diameter.

```
Ball ball1, ball2;
float gravity = 0.1;

void setup() {
  size(400, 400);

  ball1 = _____;

  ball2 = _____;
  fill(255,0,0);
}

void draw() {
  background(255);

  _____;

  _____;

  _____;

  _____;
}

class Ball {
  float x, y, xVel, yVel;
  int diam;

  Ball(float xStart, float yStart, float xV, float yV, int diameter) {
    x = xStart;
    y = yStart;
    xVel = xV;
    yVel = yV;
    diam = diameter;
  }

  void update() {
    // check if hit the ground so need to reverse the velocity
    if (y > height-diam/2) {
      yVel = -yVel;
    }
    // adjust position based on velocity
    y = y + yVel;
    x = x + xVel;

    // adjust the velocity - increasing due to gravity
    yVel = yVel + gravity; // always accelerate down

    // add some drag
    yVel = yVel*0.99;
  }

  void draw() {
    // draw the ball
    ellipse(x, y, diam, diam);
  }
}
```

21. **(2 points)** Create a class to represent rectangles. A rectangle is specified (as with drawing them in Processing) by the coordinates of its upper left corner, its width, and its height. Give your class just enough functionality to make it work with the program below. The `display()` method should simply draw the rectangle with no change in fill or stroke.

```
Rectangle[] boxes = new Rectangle[100];
void setup() {
  size(400,400);
  for (int i = 0; i < boxes.length; i++) {
    boxes[i] = new Rectangle((int)random(width), (int)random(height),
                             (int)random(10,40), (int)random(5,20));
  }
}

void draw() {
  for (int i = 0; i < boxes.length; i++) {
    fill(random(255), random(255), random(255));
    boxes[i].display();
  }
}
```