



You may use one page of hand written notes (both sides) and a dictionary.  
No i-phones, calculators nor any other type of non-organic computer.  
Time Limit: 2 Hours

1) This Java program compiles and runs. What is the output?

```
1) public class P1
2) {
3)
4)     public static void main(String[] args)
5)     {
6)         int a = 2, b = 3, c = 5;
7)         String history="";
8)         if (a*b > c)
9)         { history += "Z";
10)            if (b < a) history += "Y";
11)            else if (b < c) history += "X";
12)            else if (c < b) history += "W";
13)            history += "V";
14)        }
15)        else if (b*c > a)
16)        { history += "T";
17)            if (a > b) history += "S";
18)            else if (b > c) history += "R";
19)            else if (c > a) history += "Q";
20)            else if (c > b) history += "P";
21)        }
22)        else if ((a+b) <= c)
23)        { history += "N";
24)            if ((b - a) < c) history += "L";
25)            else if ((c - a) < b) history += "K";
26)            else if ((c - a) > b) history += "J";
27)            else if ((c - a) == b) history += "H";
28)        }
29)        history += "G";
30)        System.out.println(history);
31)    }
32) }
```



2) This Java program compiles and runs. What is the output?

```
1) public class MethodsAndLogicExpressions
2) {
3)     private static char test(int x1, int y1, int x2, int y2)
4)     { if ((x1 < 99) && (x2 < 99)) return 'W';
5)       if ((x1 > 150) && (x2 > 150)) return 'E';
6)       if ((y1 < 25) && (y2 < 25)) return 'N';
7)       if ((y1 > 75) && (y2 > 75)) return 'S';
8)       if ((x1 < 99) || (x1 > 150) || (y1 < 25) || (y1 > 75))
9)       {
10)          if ((x2 < 99) || (x2 > 150) || (y2 < 25) || (y2 > 75))
11)          { return 'O';
12)            }
13)          else return 'X';
14)        }
15)      else
16)      {
17)          if ((x2 < 99) || (x2 > 150) || (y2 < 25) || (y2 > 75))
18)          { return 'Y';
19)            }
20)          else return 'I';
21)        }
22)    }
23)
24)
25)     public static void main(String[] args)
26)     {
27)         System.out.println("1" + test( 0, 0, 10,10));
28)         System.out.println("2" + test(120,50, 130,70));
29)     }
30) }
```

3) What would be the output of `MethodsAndLogicExpressions` if lines 27 and 28 are replaced with:

```
27)     System.out.println("3" + test( 0,50, 200,50));
28)     System.out.println("4" + test(120,50, 200,50));
```



4) This Java program compiles and runs. What is the output?

```
1) public class P4
2) {
3)     public static int foo(int a, int b)
4)     {
5)         if (a > b)
6)         { int c=a; a=b; b=c;
7)         }
8)         for (int i=a; i>1; i--)
9)         {
10)            System.out.print(i + " "); // *** OUTPUT ****
11)            if ((a % i) == (b % i))
12)            {
13)                return i;
14)            }
15)        }
16)        return 1;
17)    }
18)    public static void main(String[] args)
19)    {
20)        System.out.println("==>" + foo(5, 8)); // *** OUTPUT ***
21)        System.out.println("==>" + foo(9, 5)); // *** OUTPUT ***
22)        System.out.println("==>" + foo(5, 9)); // *** OUTPUT ***
23)    }
24) }
```



5) This Java program compiles and runs. What is the output?

```
1) public class LoopNest
2) {
3)     public static void main(String[] args)
4)     {
5)         String[] animation =
6)         { "Troll", "Spider", "Rat", "Zombie"
7)         };
8)         String[] type = {"jpg", "png", "gif"};
9)
10)        for (int a=1; a<animation.length; a++)
11)        {
12)            for (int b=0; b<type.length; b++)
13)            {
14)                for (int c=2; c<4; c++)
15)                {
16)                    String str = animation[a] + c + '.' + type[b];
17)
18)                    System.out.println(str); // *** OUTPUT ***
19)                }
20)            }
21)        }
22)    }
23) }
```



6) This Java program compiles and runs. What is the output?

```
1) public class ArrayRangeScaleAndShift
2) {
3)     public static void main(String[] args)
4)     {
5)         int[] x = {2, -3, -2, 1, 3, 0};
6)         int min = -3;
7)         int max = 3;
8)         int min2 = 2;
9)         int max2 = 17;
10)
11)        for (int i=0; i<x.length; i++)
12)        {
13)            int y = min2 + (x[i]-min) * (max2-min2)/(max-min);
14)            int z = min2 + (x[i]-min) * ((max2-min2)/(max-min));
15)
16)            System.out.println(
17)                "x[" + i + "]=" + x[i]+ "--> y=" + y + ", z=" + z);
18)        }
19)    }
20) }
```



- 7) This Java program uses Picture class developed in CS-152. The program compiles, runs and displays a JFrame window with a black background. Sketch an image of what is drawn within the JFrame. This is a sketch. You need to capture the approximate shapes and patterns. You do not need to reproduce the exact measurements, nor the exact number of lines.

```
1) import java.awt.Graphics;
2) import java.awt.Color;
3)
4) public class Test
5) {
6)     public static void main(String[] args)
7)     {
8)         Picture myPic = new Picture(201, 201);
9)         Graphics canvas = myPic.getOffScreenGraphics();
10)        canvas.setColor(Color.GREEN);
11)        for (int a=0; a<100; a+=20)
12)        {
13)            int b = 100+a;
14)            int c = 100-a;
15)            canvas.drawLine(b, 2*a, b, 2*a+10);
16)            canvas.drawLine(b-5, 2*a+5, b+5, 2*a+5);
17)
18)            canvas.drawLine(c, 2*a, c, 2*a+10);
19)            canvas.drawLine(c-5, 2*a+5, c+5, 2*a+5);
20)        }
21)        myPic.repaint();
22)    }
23) }
```



- 8) This Java program uses Picture class developed in CS-152. The program compiles, runs and displays a JFrame window with a black background. Sketch an image of what is drawn within the JFrame. This is a sketch. You need to capture the approximate shapes and patterns. You do not need to reproduce the exact measurements, nor the exact number of lines.

```
1) import java.awt.Graphics;
2) import java.awt.Color;
3) public class Test
4) {
5)     public static void main(String[] args)
6)     {
7)         Picture myPic = new Picture(425, 201);
8)         Graphics canvas = myPic.getOffScreenGraphics();
9)         canvas.setColor(Color.RED);
10)        for (int n=0; n<16; n++)
11)        {
12)            int a = 20 + ((n*50) % 200);
13)            int x = n*25+20;
14)            for (int i=-2; i<=2; i++)
15)            {
16)                canvas.drawLine(x, a, x-i*5, a-10);
17)            }
18)            canvas.drawLine(x, a, x, a+10);
19)        }
20)    }
21) }
```



9) This Java program compiles and runs. What is the output?

```
1) public class Histogram
2) {
3)     public static void main(String[] args)
4)     {
5)         char[] grade =
6)         { 'C', 'B', 'C', 'C', 'B', 'C', 'C', 'A', 'D', 'B'
7)         };
8)         int[] hist = new int[5];
9)
10)        for (int i = 0; i < grade.length; i++)
11)        {
12)            int numGrade = grade[i] - 'A';
13)            hist[numGrade]++;
14)            System.out.println(i + ":"
15)                + " hist[" + numGrade + "]= " + hist[numGrade]);
16)        }
17)    }
18) }
```





10) This Java program compiles and runs. What is the output?

```
1) public class MagicSquare
2) {
3)     public static boolean isMagic(String str)
4)     {
5)         System.out.println("isMagic("+str+"):"); //<-----
6)
7)         int sum = 0;
8)         if (str.length() != 3) return false;
9)         for (int i=0; i<str.length(); i++)
10)        {
11)            char c = str.charAt(i);
12)
13)            if (!Character.isDigit(c)) return false;
14)
15)            sum = sum + Character.getNumericValue(c);
16)            if (sum > 15) return false;
17)            System.out.println("  "+c+ " , sum=" +sum); //<-----
18)
19)        }
20)        if (sum == 15) return true;
21)        return false;
22)    }
23)
24)    public static void main(String[] args)
25)    {
26)        System.out.println(isMagic("15")); //<-----
27)        System.out.println(isMagic("x39")); //<-----
28)        System.out.println(isMagic("239")); //<-----
29)        System.out.println(isMagic("987")); //<-----
30)        System.out.println(isMagic("2517")); //<-----
31)    }
32) }
```