```
7.  class J0100 {
8.  public static void main(String args[]) {
9.  System.out.println(args.length);
10. System.out.println(args[0]);   // abc
11. }
12. }
15. class J0101 {
16. public static void main(String args[]) {
18. boolean b = true;
19. System.out.println("boolean = "+b);
21. char y;
22. y = 'a';
23. System.out.println("character = "+y);
25. byte c;
26. c = 127;
27. System.out.println("byte = "+c);
29. short a;
30. a = 32767;
31. System.out.println("Short = "+a);
33. int x;
34. x = 2147483647;
35. System.out.println("Integer = "+x);
37. long b;
38. b = 9223372036854775807L;
39. System.out.println("long = "+b);
40. }
41. }
46. class J0102 {
47. public static void main(String args[]) {
```

```
49. float d;
50. d = 340000000000000000000000000000000000000f;
51. System.out.println("float = "+d);
53. double e;
54. e = 179000000000000000000000000000000000000000d;
55. System.out.println("double = "+e);
61. String z ="ThaiAll";
62. System.out.println("string = "+z);
63. System.out.println(z.substring(0,4));  // Thai
64. System.out.println(z.substring(2,5));  // aiA
65. System.out.println(z.substring(4));    // All
66. System.out.println(z.toUpperCase());   // THAIALL
67. System.out.println(z.toLowerCase());   // thaiall
68. char ar[] = new char[128];
69. ar = z.toCharArray();
70. System.out.println((char)ar[0]);      // T
71. System.out.println(ar[0]);            // T
72. System.out.println(ar[2] + ar[4]);    // 162 (97 + 65)
73. z = "1234.1";
74. int m = Integer.parseInt(z.substring(0,3)) + 5;  // 123 + 5
75. double n = Double.parseDouble(z) + 0.2;         // 1234.3
76. System.out.println(m + n);          // 128 + 1234.3 = 1362.3
77. System.out.println(Integer.toString(m) + 5);     // 1285
78. }
79. }
84. class J0201 {
85. public static void main(String args[]) {
86. int x;
87. x = 6;
```

```
88. if (x > 5) System.out.println("more than 5:" + x);
89. if (x > 5 && x < 10) System.out.println("five to ten");
90. if (x > 5 || x < 10) System.out.println("all numbers");
91. if (x > 10) {
92. System.out.print("more than 10:");
93. System.out.println(x);
94. }
95. }
96. }
101.        import java.lang.*;
102.        class J0202 {
103.        public static void main(String args[]) {
104.        int x;
105.        x = 6;
106.        if (x > 5) System.out.println("more than 5");
107.        else System.out.println("less than or equal 5");
108.        if (x > 10) System.out.println("more than 10");
109.        else { System.out.println("less than or equal 10"); }
110.        Comparable a[] = new Comparable[5];
111.        a[0] = new Integer(3);
112.        a[1] = new Integer(10);
113.        a[2] = "abc";
114.        System.out.println(a[0] + " " + a[1] + " " + a[2]);
115.        if (a[2].equals("abc")) { System.out.println("equal"); }
116.        if (a[0].compareTo(a[1]) < 0) System.out.print(a[0]); // 3
117.        if (a[1].compareTo(a[0]) > 0) System.out.print(a[0]+""+a[1]); // 310
118.        if (a[0].compareTo(a[0]) == 0) System.out.print("equal"); // equal
119.        System.out.print(a[0].compareTo(a[1])); // -1
120.        }
```

```
121.        }
122.        // :::: โปรแกรมลำดับที่ 6
125.        import java.util.Date;
126.        class J0203 {
127.        public static void main(String args[]) {
128.        byte a = (byte) (new Date().getTime() % 5);
129.        switch (a) {
130.        case 1:
131.        System.out.println("one"); break;
132.        case 2:
133.        System.out.println("two"); break;
134.        default:
135.        System.out.println("not found" + a);
136.        break;
137.        }
138.        }
139.        }
144.        class J0204 {
145.        public static void main(String args[]) {
146.        System.out.println("ASCII character :: ");
147.        for (int i=0; i<256; i++) {
148.        System.out.print((char)i + " ");
149.        // System.out.println(i);  0 - 255
150.        }
151.        String s = "thaiall";
152.        System.out.println(s + s.length());
153.        }
154.        }
```

```
159.    class J0205 {
160.    public static void main(String args[]) {
161.    System.out.println("print 1 to 10 :: ");
162.    int i;
163.    i = -5;
164.    while (i <= 5) {
165.    try {
166.    i++;
167.    System.out.println((double)5/i); //Infinity
168.    System.out.println(5/i); //catch ok
169.    }
170.    catch (ArithmeticException e) {
171.    System.out.println("may divide by zero");
172.    }
173.    }
174.    int k = 0;
175.    i = 0;
176.    while (i < 5) {
177.    System.out.print(++k);
178.    k = k + (i++);
179.    System.out.print(k--);
180.    } // 11122447711
181.    }
182.    }
187.    class J0206 {
188.    public static void main(String args[]) {
189.    System.out.println("print 1 to 10 :: ");
190.    int i;
191.    i = 1;
```

```
192.        try {
193.        do {
194.        System.out.println(i);
195.        i++;
196.        } while (i <= 10);
197.        }
198.        catch (ArrayIndexOutOfBoundsException e) {
199.        System.out.println("over index of array");
200.        }
201.        }
202.        }
206.        import java.io.*;
207.        class J0301 {
208.        public static void main(String args[]) throws IOException {
209.        char buf;
210.        buf = (char)System.in.read();
211.        System.out.println("Output is "+buf);
212.        }
213.        }
217.        import java.io.*;
218.        class J0302 {
219.        public static void main(String args[]) throws IOException {
220.        char buf1,buf2;
221.        buf1 = (char)System.in.read();
222.        buf2 = (char)System.in.read();
223.        System.out.println("Output is "+buf1+buf2);
224.        }
225.        }
229.        import java.io.*;
```

```
230.    class J0303 {
231.    public static void main(String args[]) throws IOException {
232.    System.out.println("Get until receive 0 [hidden is 13, 10]");
233.    char buf;
234.    do {
235.    buf = (char)System.in.read();
236.    System.out.println("Output is "+buf);
237.    } while (buf != '0');
238.    }
239.    }

243.    import java.io.*;
244.    class J0304 {
245.    public static void main(String args[]) throws IOException {
246.    BufferedReader stdin = new BufferedReader(new InputStreamReader(System.in));
247.    String buf;
248.    int i1,i2,i3;
249.    buf = stdin.readLine();
250.    i1 = Integer.parseInt(buf);
251.    buf = stdin.readLine();
252.    i2 = Integer.parseInt(buf);
253.    i3 = i1 + i2;
254.    System.out.println("Output is "+i1+" + "+i2+" = "+i3);
255.    }
256.    }
260.    import java.io.*;
261.    class J0305 {
262.    public static void main(String args[]) throws IOException {
263.    BufferedReader stdin = new BufferedReader(new InputStreamReader(System.in));
```

```
264.        String buf;
265.        int i;
266.        System.out.println("Get until receive 0");
267.        do {
268.        buf = stdin.readLine();
269.        i = Integer.parseInt(buf);
270.        System.out.println("Output is "+i);
271.        } while (i != 0);
272.        }
273.        }
277.        class J0401 {
278.        public static void main(String args[]) {
279.        sub1(); sub2(); sub1();
280.        }
281.        static void sub1() {
282.        System.out.print("x");
283.        }
284.        static void sub2() { System.out.print("y"); }
285.        }
289.        class J0402 {
290.        public static void main(String args[]) {
291.        int s = 0;
292.        s = sub(2,8,s);
293.        s = sub(7,3,s);
294.        s = sub(4,6,s);
295.        System.out.println("Sum = "+s);
296.        }
297.        public static int sub(int x, int y, int z) {
298.        int a = y + x + z;
```

```
299.        return (a + y + x + z);
300.        }
301.        }
305.        class J0403 {
306.        public static void main(String args[]) {
307.        int j = 3;
308.        System.out.println(doubleofnumber(j));
309.        }
310.        static int doubleofnumber(int i) {
311.        i = i * 2;
312.        return (i);
313.        }
314.        }
319.        class sub01 {
320.        void subx() {
321.        System.out.println("subx in sub01");
322.        }
323.        }
324.        class sub02 {
325.        void subx() {
326.        System.out.println("subx in sub02");
327.        }
328.        }
329.        class J0404 extends sub02 {
330.        j0404() {
331.        super.subx(); // subx in sub02
332.        this.subx(); // subx in main
333.        }
334.        public static void main(String args[]) {
```

```
335.        sub01 x = new sub01();
336.        System.out.println("main"); // main
337.        x.subx();     // subx in sub01
338.        j0404 y = new j0404();
339.        }
340.        void subx() {
341.        System.out.println("subx in main");
342.        }
343.        }
346.        class J0501 {
347.        public static void main(String args[]) {
348.        int x[] = {4,18,12};
349.        System.out.println("Amount of array = " + x.length);
350.        for (int i = 0; i < x.length; i++) {
351.        System.out.println("element "+i+" = "+x[i]);
352.        }
353.        }
354.        }
358.        class J0502 {
359.        public static void main(String args[]) {
360.        String a[][] = new String[2][3];
361.        a[0][0] = "101";
362.        a[0][1] = "102";
363.        a[0][2] = "103";
364.        int i = 0;
365.        a[1][i++] = "tom";  // 1,0
366.        a[1][i++] = "dang"; // 1,1
367.        a[1][i++] = "boy";  // 1,2
368.        for (i = 0; i < a[0].length; i++) {
```

```
369.        System.out.println("element of 0,"+i+" = "+a[0][i]);
370.        }
371.        for (i = 0; i < a[1].length; i++) {
372.        System.out.println("element of 1,"+i+" = "+a[1][i]);
373.        }
374.        }
375.        }
380.        import java.io.*;
381.        class J0601 {
382.        public static void main (String args[]) throws IOException {
383.        File f = new File("j0601.java");
384.        System.out.println("getName: "+f.getName());
385.        System.out.println("getPath: "+f.getPath());
386.        System.out.println("getAbsolutePath: "+f.getAbsolutePath());
387.        System.out.println("exists: "+f.exists());
388.        System.out.println("isFile: "+f.isFile());
389.        System.out.println("isDirectory: "+f.isDirectory());
390.        System.out.println("canWrite: "+f.canWrite());
391.        System.out.println("canRead: "+f.canRead());
392.        System.out.println("length: "+f.length());
393.        File file = new File("hello.txt");
394.        boolean success = file.createNewFile();
395.        File file2 = new File("hello.java");
396.        success = file.renameTo(file2);
397.        File b = new File("c:/");
398.        success = file2.renameTo(new File(b, file2.getName()));
399.        success = (new File("hello.java")).delete();
400.        System.out.println(success); // false
401.        }
```

```
402.        }
406.        import java.io.*;
407.        class J0602 {
408.        public static void main (String args[]) {
409.        File d = new File(args[0]);
410.        String n[] = d.list();
411.        for (int i = 0; i<n.length; i++) {
412.        File f = new File(args[0] + '/' + n[i]);
413.        System.out.println(i+" : "+n[i]+" Size="+f.length());
414.        }
415.        System.out.println("directory: "+d.getPath());
416.        }
417.        }
420.        import java.io.*;
421.        class J0603 {
422.        public static void main (String args[]) throws IOException {
423.        int n = 0;
424.        byte b[] = new byte[128];
425.        FileInputStream fin = new FileInputStream("j0603.java");
426.        while ((n = fin.read(b)) != -1) {
427.        for(int i=0;i<n;i++) System.out.print((char)b[i]);
428.        }
429.        System.out.println(n = fin.read(b)); // -1
430.        fin.close();
431.        }
432.        }
436.        import java.io.*;
437.        class J0604 {
438.        public static void main (String args[]) throws IOException {
```

```
439.        FileOutputStream fout = new FileOutputStream("tmp.txt");
440.        for(int i=0;i<256;i++) {
441.        fout.write(i);
442.        }
443.        fout.close();
444.        }
445.        }
448.        import java.io.*;
449.        class J0605 {
450.        public static void main (String args[]) throws IOException {
451.        FileOutputStream fout = new FileOutputStream("tmp.txt");
452.        for(int i=1;i<=10;i++) {
453.        fout.write(i+47);
454.        fout.write(13);
455.        fout.write(10);
456.        }
457.        fout.close();
458.        }
459.        }
464.        import java.io.*;
465.        class J0606 {
466.        public static void main (String args[]) throws IOException {
467.        int i = 0, n = 0;
468.        char b[] = new char[1];
469.        FileReader fin = new FileReader("tmp.txt");
470.        while ((n =  fin.read(b)) != -1) {
471.        System.out.println(i+" : "+b[0]);
472.        i = i + 1;
473.        }
```

```
474.        fin.close();
475.        }
476.        }
481.        import java.io.*;
482.        class J0607 {
483.        public static void main (String args[]) throws IOException {
484.        int i = 1, n = 0;
485.        char b[] = new char[16];
486.        FileReader fin = new FileReader("tmp.txt");
487.        while ((n =  fin.read(b)) != -1) {
488.        System.out.print((i-1)*16 + " - " + (i*16-1) + ":");
489.        System.out.print(b[0]+b[1]+b[2]+b[3]+b[4]+b[5]+b[6]+b[7]+b[8]);
490.        System.out.println(b[9]+b[10]+b[11]+b[12]+b[13]+b[14]+b[15]);
491.        i = i + 1;
492.        }
493.        fin.close();
494.        }
495.        }
499.        import java.io.*;
500.        class J0608 {
501.        public static void main (String args[]) throws IOException {
502.        int i = 1;
503.        String b;
504.        FileReader fin = new FileReader("data.txt");
505.        BufferedReader bin = new BufferedReader (fin);
506.        // System.out.println(b = bin.readLine()); // output is b
507.        while ((b =  bin.readLine()) != null) {
508.        System.out.println(i + " : " +b);
509.        i = i + 1;
```

```
510.        }
511.        System.out.println(b = bin.readLine()); // null
512.        fin.close();
513.        }
514.        }
520.        import java.io.*;
521.        class J0701 {
522.        public static void main (String args[]) throws IOException {
523.        int i = 1;
524.        int tot = 0;
525.        String b;
526.        String[] fields;
527.        String patternStr = ",";
528.        FileReader fin = new FileReader("data.txt");
529.        BufferedReader bin = new BufferedReader (fin);
530.        while ((b =  bin.readLine()) != null) {
531.        fields = b.split(patternStr);
532.        System.out.println(i + " : " + fields[0]);
533.        System.out.println("Name : " + fields[1]);
534.        System.out.println("Salary : " + fields[2]);
535.        System.out.println("Status : " + fields[3]);
536.        tot = tot + Integer.parseInt(fields[2]);
537.        i = i + 1;
538.        }
539.        System.out.println("Total : " + tot);
540.        fin.close();
541.        }
542.        }
548.        import java.io.*;
```

```
549.    import java.lang.*;
550.    class J0702 {
551.    public static void main (String args[]) throws IOException {
552.    int i = 1;
553.    String b;
554.    String[] fields;
555.    String patternStr = ",";
556.    FileReader fin = new FileReader("data.txt");
557.    BufferedReader bin = new BufferedReader (fin);
558.    FileOutputStream fout = new FileOutputStream("data.htm");
559.    BufferedOutputStream bout = new BufferedOutputStream(fout);
560.    PrintStream pout = new PrintStream(bout);
561.    pout.println("<body bgcolor=yellow><table border=1 width=100%>");
562.    while ((b =  bin.readLine()) != null) {
563.    fields = b.split(patternStr);
564.    pout.println("<tr>");
565.    pout.println("<td>"+i+"</td>");
566.    pout.println("<td>"+"ID = " + fields[0]+"</td>");
567.    pout.println("<td>"+"Name = " + fields[1]+"</td>");
568.    pout.println("<td>"+"Salary = " + fields[2]+"</td>");
569.    pout.println("<td>"+"Status = " + fields[3]+"</td>");
570.    pout.println("</tr>");
571.    i = i + 1;
572.    }
573.    pout.println("</table></body>");
574.    fin.close();
575.    pout.close();
576.    }
577.    }
```

```
581.    import java.io.*;
582.    class J0703 {
583.    public static void main (String args[]) throws IOException {
584.    int i = 0,d;
585.    String b;
586.    String[] fields;
587.    String[] recs = {"","","",""};
588.    String patternStr = ",";
590.    FileReader fin = new FileReader("data.txt");
591.    BufferedReader bin = new BufferedReader (fin);
593.    while ((b =  bin.readLine()) != null) {
594.    recs[i] = b;
595.    i = i + 1;
596.    }
597.    fin.close();
599.    FileOutputStream fout = new FileOutputStream("data.htm");
600.    BufferedOutputStream bout = new BufferedOutputStream(fout);
601.    PrintStream pout = new PrintStream(bout);
602.    for(int j=0;j<i;j++) {
603.    fields = recs[j].split(patternStr);
604.    pout.print(fields[0]+","+fields[1]+",");
606.    d = Integer.valueOf(fields[2]).intValue() + 100;
607.    pout.print(d);
608.    pout.println(","+fields[3]);
609.    }
610.    pout.close();
611.    }
612.    }
617.    import java.io.*;
```

```
618.    class J0801 {
619.    public static void main (String args[]) throws IOException {
620.    int found=0;
621.    char buf;
622.    String b,g = "";
623.    String[] fields;
624.    String patternStr = ",";
625.    System.out.println("Wait id and end character with [x]");
626.    buf = (char)System.in.read();
627.    while (buf != 'x') {
628.    g = g + buf;
629.    buf = (char)System.in.read();
630.    }
631.    FileReader fin = new FileReader("data.txt");
632.    BufferedReader bin = new BufferedReader (fin);
633.    while ((b =  bin.readLine()) != null) {
634.    fields = b.split(patternStr);
635.    if (fields[0].equals(g)) {
636.    System.out.println(fields[1]);
637.    found = 1;
638.    }
639.    }
640.    if (found == 0) System.out.println("Not found");
641.    fin.close();
642.    }
643.    }
647.    import java.io.*;
648.    class J0802 {
649.    public static void main (String args[]) throws IOException {
```

```
650.        int found=0;
651.        String b,g = "";
652.        String[] fields;
653.        System.out.println("Wait string and enter");
654.        BufferedReader stdin = new BufferedReader(new InputStreamReader(System.in));
655.        g = stdin.readLine();
656.        String patternStr = g;
657.        FileReader fin = new FileReader("data.txt");
658.        BufferedReader bin = new BufferedReader (fin);
659.        while ((b =  bin.readLine()) != null) {
660.        fields = b.split(patternStr);
661.        if (fields.length > 1) {
662.        fields = b.split(",");
663.        System.out.println(fields[0] + fields[1] + fields[2] + fields[3]);
664.        found = 1;
665.        }
666.        }
667.        if (found == 0) System.out.println("Not found");
668.        fin.close();
669.        }
670.        }
681.        import java.io.*;
682.        class J0901 {
683.        public static void main (String args[]) throws IOException {
684.        int i = 0,t1,t2;
685.        String b,status;
687.        String fields[];
688.        String[] recs1 = new String[10];
689.        String[] recs2 = {"A,Active","R,Retire"};
```

```
690.        String patternStr = ",";
692.        FileReader fin = new FileReader("data.txt");
693.        BufferedReader bin = new BufferedReader (fin);
694.        while ((b =  bin.readLine()) != null) {
695.        recs1[i] = b;
696.        i = i + 1;
697.        }
698.        fin.close();
699.        t1 = i;
700.        t2 = recs2.length;
702.        for(int j=0;j<t1;j++) {
703.        fields = recs1[j].split(patternStr);
704.        System.out.print(fields[0] + fields[1] + fields[2]+fields[3]);
705.        status = fields[3];
706.        for(int k=0;k<t2;k++) {
707.        fields = recs2[k].split(patternStr);
708.        if (fields[0].equals(status)) {
709.        System.out.println(fields[1]);
710.        }
711.        }
712.        }
713.        }
714.        }
720.        import java.io.*;
721.        class J0902 {
722.        public static void main (String args[]) throws IOException {
723.        int i = 0,t1,t2;
724.        String b,status;
725.        String[] fields;
```

```
726.        String[] recs1 = {"","","","","","",""};
727.        String[] recs2 = new String[2];
728.        FileReader fin = new FileReader("data.txt");
729.        BufferedReader bin = new BufferedReader (fin);
730.        while ((b =  bin.readLine()) != null) {
731.        recs1[i] = b;
732.        i = i + 1;
733.        }
734.        fin.close();
735.        t1 = i;
736.        i = 0;
738.        FileReader fin2 = new FileReader("datas.txt");
739.        BufferedReader bin2 = new BufferedReader (fin2);
740.        while ((b =  bin2.readLine()) != null) {
741.        recs2[i] = b;
742.        i = i + 1;
743.        }
744.        fin2.close();
745.        t2 = i;
747.        for(int j=0;j<t1;j++) {
748.        fields = recs1[j].split(",");
749.        System.out.print(fields[0] + fields[1] + fields[2]+fields[3]);
750.        status = fields[3];
751.        for(int k=0;k<t2;k++) {
752.        fields = recs2[k].split(",");
753.        if (fields[0].equals(status)) {
754.        System.out.println(fields[1]);
755.        }
756.        }
```

```
757.        }
758.        }
759.        }
764.        class J1001 {
765.        public static void main (String args[]) {
766.        int tmp,x[] = {5,6,1,2,9,12,9,3};
767.        for(int i=1;i<x.length;i++) {
768.        for(int j=x.length-1;j>=i;j--) {
769.        if(x[j-1] > x[j]) {
770.        tmp = x[j];
771.        x[j] = x[j-1];
772.        x[j-1] = tmp;
773.        }
774.        }
775.        }
776.        for(int i=0;i<x.length;i++) {
777.        System.out.println(x[i]);
778.        }
779.        }
780.        }
785.        import java.lang.*;
786.        class J1002 {
787.        public static void main (String args[]) {
788.        String tmp,x[] = {"ac","abc","adb","a","aa","acd","a a","a d"};
789.        System.out.println("Before sorting");
790.        prtlist(x);
791.        for(int i=1;i<x.length;i++) {
792.        for(int j=x.length-1;j>=i;j--) {
793.        if(x[j-1].compareTo(x[j])>0) {
```

```
794.        tmp = x[j];
795.        x[j] = x[j-1];
796.        x[j-1] = tmp;
797.        }
798.        }
799.        }
800.        System.out.println("After sorting");
801.        prtlist(x);
802.        }
803.        public static void prtlist(String[] x) {
804.        for(int i=0;i<x.length;i++) {
805.        System.out.println(x[i]);
806.        }
807.        }
808.        }
815.        import java.applet.*;
816.        import java.awt.*;
817.        public class J1101 extends java.applet.Applet {
818.        public void paint(Graphics g) {
819.        g.setColor(new Color(240,240,240));
820.        g.drawString("test",10,20);
821.        }
822.        }
828.        import java.applet.*;
829.        import java.awt.*;
830.        public class J1102 extends Applet {
831.        int i,j;
832.        String istr,p;
833.        public void init() {
```

```
834.        setBackground(Color.yellow);
835.        p = getParameter("x");
836.        }
837.        public void paint(Graphics g) {
838.        g.setColor(Color.black);
839.        g.drawString(p,0,10);
840.        i = 1;
841.        while (i <= 10) {
842.        j = 10 * i;
843.        istr= Integer.toString(i);
844.        g.drawString(istr,72,j); // column = 1 inch
845.        i++;
846.        }
847.        }
848.        }
853.        import java.applet.*;
854.        import java.awt.*;
855.        public class J1103 extends Applet implements Runnable{
856.        Thread timer;
857.        int row = 10;
858.        public void paint(Graphics g) {
859.        row = row + 2;
860.        g.drawLine(5,row,30,row);
861.        }
862.        public void start() {
863.        timer = new Thread(this);
864.        timer.start(); // start clock
865.        }
866.        public void run() {
```

```
867.        Thread me = Thread.currentThread();
868.        while (timer == me) {
869.        try {
871.        Thread.currentThread().sleep(1000);
872.        } catch (InterruptedException e) { }
873.        repaint();
874.        }
875.        }
876.        }
881.        import java.applet.*;
882.        import java.awt.*;
883.        public class J1104 extends Applet {
884.        Image img;
885.        public void init() {
886.        setBackground(Color.green);
887.        img = getImage(getDocumentBase(),"x.gif");
888.        }
889.        public void paint(Graphics g) {
890.        g.setColor(Color.black);
891.        g.drawLine(5,10,30,40);
892.        g.drawRect(50,50,80,80);
893.        g.drawOval(50,50,20,30);
894.        g.setColor(Color.white);
895.        g.fillOval(50,50,20,30); // backgound is white
896.        g.setColor(Color.red);
897.        g.drawArc(40,30,55,55,0,120);
898.        int[] x={0,80,100,5,10};
899.        int[] y={0,50,80,80,30};
900.        g.drawPolygon(x,y,5);
```

```
901.        g.drawImage(img, 0, 200, this);
902.        }
903.        }
907.        import java.applet.*;
908.        import java.awt.*;
909.        import java.awt.event.*;
910.        public class J1105 extends Applet implements ActionListener {
911.        Button b1 = new Button("1");
912.        Label l1 = new Label("Hello");
913.        TextField t1 = new TextField("1");
914.        int row = 10;
915.        public void paint(Graphics g) {
916.        row = row + 10;
917.        g.drawLine(5,row,30,row);
918.        }
919.        public void init() {
920.        setBackground(Color.red);
921.        add(l1);
922.        add(b1);
923.        add(t1);
924.        t1.addActionListener(this);
925.        b1.addActionListener(this);
926.        }
927.        public void actionPerformed(ActionEvent e) {
928.        int intb1 = Integer.parseInt(e.getActionCommand());
929.        intb1 = intb1 + 1;
930.        String s = Integer.toString(intb1);
931.        l1.setText(s);
932.        b1.setLabel(s);
```

```
933.        t1.setText(s);
934.        repaint();
935.        }
936.        }
937.
941.        import java.io.*;
942.        class J1201 {
943.        public static void main(String args[]) throws IOException {
944.        int buf=49;
945.        while (buf != 51) {
946.        if (buf >= 49 && buf <= 51) {
947.        System.out.println("What is your option?");
948.        System.out.println("1. print 1 to 10");
949.        System.out.println("2. print 'ok'");
950.        System.out.println("3. exit");
951.        }
953.        buf = System.in.read();
954.        switch (buf) {
955.        case 49: // character 1
956.        for (int i=1;i<=10;i++) {
957.        System.out.println(i);
958.        }
959.        break;
960.        case 50: // character 2
961.        System.out.println("ok");
962.        break;
963.        case 51: break; // character 3
964.        case 13: break;
965.        case 10: break;
```

```
966.        default:
967.        System.out.println("Nothing to do");
968.        break;
969.        }
970.        }
971.        System.out.println("See you again");
972.        }
973.        }
977.        import java.io.*;
978.        class J1202 {
979.        public static void main(String args[]) throws IOException {
980.        BufferedReader stdin = new BufferedReader(new InputStreamReader(System.in));
981.        String buf=" ";
982.        while (!buf.equals("3")) {
983.        System.out.println("What is your option?");
984.        System.out.println("1. print 1 to 10");
985.        System.out.println("2. print 'ok'");
986.        System.out.println("3. exit");
987.        buf = stdin.readLine();
988.        if (buf.equals("1"))
989.        for (int i=1;i<=10;i++) System.out.println(i);
990.        if (buf.equals("2")) System.out.println("ok");
991.        }
992.        System.out.println("See you again");
993.        }
994.        }
998.        import java.io.*;
999.        class J1203 {
1000.       public static void main(String args[]) throws IOException {
```

```
1001.       BufferedReader stdin = new BufferedReader(new InputStreamReader(System.in));
1002.       String buf=" ";
1003.       while (!buf.equals("3")) {
1004.       System.out.println("What is your option?");
1005.       System.out.println("1. print 1 to 10");
1006.       System.out.println("2. print 'ok'");
1007.       System.out.println("3. exit");
1008.       buf = stdin.readLine();
1009.       if (buf.equals("1")) oho1();
1010.       if (buf.equals("2")) { oho2(); }
1011.       }
1012.       System.out.println("See you again");
1013.       }
1014.       public static void oho1() {
1015.       for (int i=1;i<=10;i++) {
1016.       System.out.println(i);
1017.       }
1018.       }
1019.       public static void oho2() {
1020.       System.out.println("ok");
1021.       }
1022.       }
1024.       import java.io.*;
1025.       class Pollweb {
1026.       public static void main (String args[]) throws IOException {
1027.       int i=0;
1028.       int questionhave = 14;
1029.       int q[] = new int[questionhave];
1030.       String b;
```

```
1031.       String[] fields;
1032.       String patternStr = ",";
1033.       FileReader fin = new FileReader("pollweb.txt");
1034.       BufferedReader bin = new BufferedReader (fin);
1035.       while ((b =  bin.readLine()) != null) {
1036.       fields = b.split(patternStr);
1037.       for (int j=1;j<=questionhave-1;j++)
1038.       q[j]+= Integer.parseInt(fields[j]);
1039.       i = i + 1;
1040.       }
1041.       System.out.println("Total questions: " + i);
1042.       for (int j=1;j<=questionhave-1;j++)
1043.       System.out.println(j+":"+q[j]+" | "+(q[j] * 100 / i)+"%");
1044.       fin.close();
1045.       }
1046.       }
1048.       class Hello1 {
1049.       public static void main(String args[]) {
1050.       System.out.println("hello");
1051.       }
1052.       }
1054.       import java.lang.*;
1055.       import java.applet.*;
1056.       import java.awt.Graphics;
1057.       public class Hello2 extends java.applet.Applet {
1058.       public void paint(Graphics g){
1059.       g.drawString("hello",10,10);
1060.       }
1061.       }
```

```
1063.       class Pyramid01 {
1064.       public static void main(String args[]) {
1065.       int k = 4;
1066.       for (int i=1;i<=k;i++) {
1067.       for (int j=2;j<=i;j++) { System.out.print(" "); }
1068.       System.out.print(i+""+i);
1069.       for (int j=k;j>=(i+1);j--) { System.out.print("**"); }
1070.       System.out.println(i+""+i);
1071.       } } }
1073.       class Pyramid02 {
1074.       public static void main(String args[]) {
1075.       int k = 4;
1076.       for (int i=1;i<=k;i++) {
1077.       for (int j=i;j<=(i+2);j++) { System.out.print(j); }
1078.       for (int j=1;j<=(2+i);j++) { System.out.print("*"); }
1079.       System.out.println();
1080.       } } }
1082.       class Pyramid03 {
1083.       public static void main(String args[])   {
1084.       int k = 4;
1085.       for (int i=1;i<=k;i++) {
1086.       System.out.print(i+""+(i+4));
1087.       for (int j=1;j<=(4+i);j++) {
1088.       System.out.print("*");
1089.       }
1090.       System.out.println();
1091.       } } }
1093.       class Pyramid04 {
1094.       public static void main(String args[]) {
```

```
1095.       int k = 4;
1096.       for (int i=1;i<=k;i++) {
1097.       for (int j=1;j<=i;j++) { System.out.print("*"); }
1098.       for (int j=i;j>=2;j--) { System.out.print(j); }
1099.       for (int j=1;j<=i;j++) { System.out.print(j); }
1100.       System.out.println();
1101.       } } }
```