Expressions and Operators

UNARY

IncDec.java

1.

```
public class IncDec
{   public static void main (String args[])
    {   int x = 8, y = 13;
        System.out.println("x = " + x);
        System.out.println("y = " + y);
        System.out.println("++x = " + ++x);
        System.out.println("y++ = " + y++);
        System.out.println("x = " + x);
        System.out.println("y = " + y);
    }
}
```

Negation.java

2.

```
public class Negation
{   public static void main (String args[])
    {   int x = 8;
        System.out.println("x = " + x);
        int y = -x;
        System.out.println("y = " + y);
    }
}
```

BINARY

Arithmetic.java

3.

```
public class Arithmetic
{    public static void main (String args[])
{    int x = 17, y = 5;
     System.out.println("x = " + x);
     System.out.println("y = " + y);
     System.out.println("x + y = " + (x + y));
     System.out.println("x - y = " + (x - y));
     System.out.println("x * y = " + (x * y));
     System.out.println("x / y = " + (x / y));
     System.out.println("x % y = " + (x % y));
}
}
```

Bitwise.java

4.

```
public class Bitwise
{    public static void main (String args[])
    {   int x = 5, y = 6;
        System.out.println("x = " + x);
        System.out.println("y = " + y);
        System.out.println("x & y = " + (x & y));
        System.out.println("x | y = " + (x | y));
        System.out.println("x ^ y = " + (x ^ y));
    }
}
```

Shift.java

5.

```
public class Shift
{   public static void main (String args[])
    {   int x = 7;
        System.out.println("x = " + x);
        System.out.println("x >> 2 = " + (x >> 2));
        System.out.println("x << 1 = " + (x << 1));
        System.out.println("x >>> 1 = " + (x >>> 1));
    }
}
```

FloatMath.java

6.

```
public class FloatMath
{   public static void main (String args[])
   {   float x = 23.5F, y = 7.3F;
       System.out.println("x = " + x);
       System.out.println("y = " + y);
       System.out.println("x + y = " + (x + y));
       System.out.println("x - y = " + (x - y));
       System.out.println("x * y = " + (x * y));
       System.out.println("x / y = " + (x / y));
       System.out.println("x % y = " + (x % y));
   }
}
```

RELATIONAL

Relational.java

7.

```
public class Relational
{   public static void main (String args[])
   {  int x = 7, y = 11, z = 11;
       System.out.println("x = " + x);
       System.out.println("y = " + y);
       System.out.println("z = " + z);
       System.out.println("x < y = " + (x < y));
       System.out.println("x > z = " + (x > z));
       System.out.println("y <= z = " + (y <= z));
       System.out.println("x >= y = " + (x >= y));
       System.out.println("y == z = " + (y == z));
       System.out.println("x != y = " + (x != z));
   }
}
```

FLOATING-POINT OPERATORS

FloatMath.java

8.

```
public class FloatMath
{   public static void main (String args[])
   {  float x = 23.5F, y = 7.3F;
       System.out.println("x = " + x);
       System.out.println("y = " + y);
       System.out.println("x + y = " + (x + y));
       System.out.println("x - y = " + (x - y));
       System.out.println("x * y = " + (x * y));
       System.out.println("x / y = " + (x / y));
       System.out.println("x % y = " + (x % y));
   }
}
```

BOOLEAN OPERATORS

Conditional.java

9.

```
public class Conditional
{   public static void main (String args[])
   {  int x = 0;
       boolean isEven = false;
       System.out.println("x = " + x);
       x = isEven ? 4 : 7;
       System.out.println("x = " + x);
   }
}
```

STRING OPERATORS

Concatenation.java

10.

```
public class Concatenation
{   public static void main (String args[])
   {   String firstHalf = "What " + "did ";
       String secondHalf = "you " + "say?";
       System.out.println(firstHalf + secondHalf);
   }
}
```

Control Structures

IF – ELSE

IfElseName.java

11.

```
public class IfElseName
{   public static void main (String args[])
    {  char firstInitial = (char)-1;
       System.out.println("Enter your first initial:");
       try { firstInitial = (char)System.in.read();}
       catch (Exception e) {System.out.println("Error: " + e.toString());}
       if (firstInitial == -1)
          System.out.println("Now what kind of name is that?");

       else if (firstInitial == 'j')
          System.out.println("Your name must be Jules!");
       else if (firstInitial=='v')
          System.out.println("Your name must be Vincent!");
       else if (firstInitial=='z')
          System.out.println("Your name must be Zed!");
       else
          System.out.println("I can't figure out your name!");
    }
}
```

Jika yang di-input adalah "Z"

SWITCH

SwitchName1.java

12.

```
public class SwitchName1
{   public static void main (String args[])
    {   char firstInitial = (char)-1;
        System.out.println("Enter your first initial:");
        try { firstInitial = (char)System.in.read(); }
        catch (Exception e) {System.out.println("Error: " + e.toString());}
        switch(firstInitial) {
           case (char)-1:  System.out.println("Now what kind of name is that?");
           case 'j': System.out.println("Your name must be Jules!");
           case 'v': System.out.println("Your name must be Vincent!");
           case 'z': System.out.println("Your name must be Zed!");
           default: System.out.println("I can't figure out your name!");
        }
    }
}
```

SwitchName2.java

13.

```
public class SwitchName2
{
   public static void main (String args[])
   {
      char firstInitial = (char)-1;
      System.out.println("Enter your first initial:");
      try  {firstInitial = (char)System.in.read();}
      catch (Exception e) {System.out.println("Error: " + e.toString());}
      switch(firstInitial)
      {
         case (char)-1: System.out.println("Now what kind of name is that?");  break;
         case 'j':  System.out.println("Your name must be Jules!"); break;
         case 'v': System.out.println("Your name must be Vincent!"); break;
         case 'z': System.out.println("Your name must be Zed!"); break;
         default: System.out.println("I can't figure out your name!");
      }
   }
}
```

LOOPS / FOR

ForCoun.java

14.

```
public class ForCount
{   public static void main (String args[])
    {  char input = (char)-1;
       int  numToCount;
       System.out.println("Enter a number to count to between 0 and 10:");
       try { input = (char) System.in.read();}
       catch (Exception e){System.out.println("Error: " + e.toString());}
       numToCount = Character.digit(input, 10);
       if ((numToCount > 0) && (numToCount < 10)) {
          for (int i = 1; i <= numToCount; i++)
             System.out.println(i);
       } else
          System.out.println("That number was not between 0 and 10!");
    }
}
```

WHILE

WhileCount.java

15.

```
public class WhileCount
{    public static void main (String args[])
     {  char input = (char)-1;
        int  numToCount;
        System.out.println("Enter a number to count to between 0 and 10:");
        try { input = (char)System.in.read();}
        catch (Exception e) { System.out.println("Error: " + e.toString());}
        numToCount = Character.digit(input, 10);
        if ((numToCount > 0) && (numToCount < 10)) {
           int i = 1;
           while (i <= numToCount) {
              System.out.println(i);
              i++;
           }
        }
        else
           System.out.println("That number was not between 0 and 10!");
     }
}
```

BREAK AND CONTINUE

BreakLoop.java

16.

```
public class BreakLoop
{   public static void main (String args[])
    {   int i = 0;
        do
        {   System.out.println("I'm stuck!");
            i++;
            if (i > 100)
               break;
        }
        while (true);
    }
}
```

Blocks and Scope

HowdyWorld.java

17.

```
public class HowdyWorld
{  public static void main (String args[])
   {  int i;
      printMessage();
   }
   public static void printMessage ()
   {  int j;
      System.out.println("Howdy, World!");
   }
}
```

ARRAY

arr02.java

18.

```
public class arr02
{
   public static void main (String args[])
   {

      int myarr[] = new int [100];  /*mendefinisikan array satu dimensi kosong sebanyak 100*/
      int nilai=1;                  /*inisialisasi nilai awal variabel nilai=1*/
      for (int i=0 ; i < 100; i++)  /*loop sampai 100 untuk mengisi myarr secara otomatis*/
      {
         myarr[i]=nilai++;          /*mengisi myarr yang ke-i dengan nilai+1*/
      }
      for (int i=0 ; i < 100; i++)  /*loop sampai hitungan 100 untuk mencetak myarr */
      {
         System.out.println("Array ["+i+"] = "+myarr[i]);    /*cetak myarr yang ke-i */
      }
   }
}
```

The utilities package

BITSET

/* BitSet1.java */

```
import java.io.DataInputStream;
import java.util.BitSet;

class BitSet1
{   public static void main(String args[])
    throws java.io.IOException
    {   DataInputStream dis=new DataInputStream(System.in);
        String bitstring;
        BitSet set1,set2,set3;
        set1=new BitSet();
        set2=new BitSet();
        System.out.println("Bit sequence #1:");        // Get the first bit sequence and store it
        bitstring=dis.readLine();
        for (short i=0;i<bitstring.length();i++){
            if (bitstring.charAt(i)=='1')
                set1.set(i);
            else
                set1.clear(i);
        }
        System.out.println("Bit sequence #2:");        // Get the second bit sequence and store it
        bitstring=dis.readLine();
        for (short i=0;i<bitstring.length();i++){
            if (bitstring.charAt(i)=='1')
                set2.set(i);
            else
                set2.clear(i);
        }
        System.out.println("BitSet #1: "+set1);
        System.out.println("BitSet #2: "+set2);
        set3=(BitSet)set1.clone();                      // Test the AND operation
        set3.and(set2);
        System.out.println("set1 AND set2: "+set3);
        set3=(BitSet)set1.clone();                      // Test the OR operation
        set3.or(set2);
        System.out.println("set1 OR set2: "+set3);
        set3=(BitSet)set1.clone();                      // Test the XOR operation
        set3.xor(set2);
        System.out.println("set1 XOR set2: "+set3);
    }
}
```

DATE

```
/* Date1.java */
import java.util.Date;
public class Date1
{   public static void main (String args[])
    {   Date today=new Date();
        System.out.println("Today is "+today.toLocaleString()+" ("+today.toGMTString()+")");
        Date birthday=new Date(89,10,14,8,30,00);
        System.out.println("My birthday is "+birthday.toString()+" ("+birthday.toGMTString()+")");
        Date anniversary=new Date("Jun 21, 1986");
        System.out.println("My anniversary is "+anniversary+" ("+anniversary.toGMTString()+")");
    }
}
```

RANDOM

```
/*Random1.java*/
import java.lang.Math;
import java.util.Date;
import java.util.Random;

class Random1
{   public static void main(String args[])
    throws java.io.IOException
    {   int count=6;
        Random randGen=new Random();
        System.out.println("Uniform Random Integers");
        for (int i=0;i<count;i++)
            System.out.print(randGen.nextInt()+" ");
        System.out.println("\n");
        System.out.println("Uniform Random Floats");
        for (int i=0;i<count;i++)
            System.out.print(randGen.nextFloat()+" ");
        System.out.println("\n");
        System.out.println("Gaussian Random Floats");
        for (int i=0;i<count;i++)
            System.out.print(randGen.nextGaussian()+" ");
        System.out.println("\n");
        System.out.println("Uniform Random Integers [1,6]");
        for (int i=0;i<count;i++)
            System.out.print((Math.abs(randGen.nextInt())%6+1)+" ");
        System.out.println("\n");
    }
}
```

STRINGTOKENIZER

/* StringTokenizer1.java */

```
import java.io.DataInputStream;
import java.util.StringTokenizer;

class StringTokenizer1
{   public static void main(String args[])
    throws java.io.IOException
    {   DataInputStream dis=new DataInputStream(System.in);
        System.out.println("Enter a sentence: ");
        String s=dis.readLine();
        StringTokenizer st=new StringTokenizer(s);
        while (st.hasMoreTokens())
            System.out.println(st.nextToken());
    }
}
```

STACK

/* Stack1.java */

```
import java.io.DataInputStream;
import java.util.Stack;
import java.util.StringTokenizer;
class Stack1 {
    public static void main(String args[])
        throws java.io.IOException
    {
        DataInputStream dis=new DataInputStream(System.in);
        System.out.println("Enter a sentence: ");
        String s=dis.readLine();
        StringTokenizer st=new StringTokenizer(s);
        Stack stack=new Stack();
        while (st.hasMoreTokens())
            stack.push(st.nextToken());
        while (!stack.empty())
            System.out.print((String)stack.pop()+" ");
        System.out.println();
    }
}
```

HASHTABLE

```
/* Hashtable1.java */
import java.io.DataInputStream;
import java.lang.Integer;
import java.lang.Math;
import java.util.Random;
import java.util.Hashtable;

class Hashtable1
{    public static void main(String args[])
     throws java.io.IOException
     {  DataInputStream dis=new DataInputStream(System.in);
        int numElements=10;
        String keys[] ={"Red","Green","Blue","Cyan","Magenta","Yellow","Black","Orange","Purple","White"};
        Hashtable ht;
        Random randGen=new Random();
        ht=new Hashtable(numElements*2);
        for (int i=0;i<numElements;i++)
           ht.put(keys[i],new Integer(Math.abs(randGen.nextInt())%numElements));
        System.out.println(ht.toString());
        String keyValue;
        System.out.println("Which key to find? ");
        keyValue=dis.readLine();
        Integer value=(Integer)ht.get(keyValue);
        if (value!=null)
           System.out.println(keyValue+" = "+value);
     }
}
```

PROPERTIES

```
/* Properties1.java */
import java.io.DataInputStream;
import java.lang.Integer;
import java.util.Properties;
class Properties1
{   public static void main(String args[])
    throws java.io.IOException
    {  int numElements=4;
       String defaultNames[]={"Red","Green","Blue","Purple"};
       int defaultValues[]={1,2,3,4};
       String userNames[]={"Red","Yellow","Orange","Blue"};
       int userValues[]={100,200,300,400};
       DataInputStream dis=new DataInputStream(System.in);
       Properties defaultProps=new Properties();
       Properties userProps=new Properties(defaultProps);
       for (int i=0;i<numElements;i++){
          defaultProps.put(defaultNames[i],Integer.toString(defaultValues[i]));
          userProps.put(userNames[i],Integer.toString(userValues[i]));
       }
       System.out.println("Default Properties");
       defaultProps.list(System.out);
       System.out.println("\nUser Defined Properties");
       userProps.list(System.out);
       String keyValue;
       System.out.println("\nWhich property to find? ");
       keyValue=dis.readLine();
       System.out.println("Property '"+keyValue+"' is '"+userProps.getProperty(keyValue)+"'");
    }
}
```

The I/O package

INPUT STREAM CLASSES

THE SYSTEM.IN OBJECT

/* ReadKeys1.java */

```
class ReadKeys1 {
   public static void main (String args[]) {
      StringBuffer s = new StringBuffer();
      char c;
      try {
         while ((c = (char)System.in.read()) != '\n') {
         s.append(c);
      }
   }
   catch (Exception e) {
      System.out.println("Error: " + e.toString());
   }
   System.out.println(s);
  }
}
```

/* ReadKeys2.java */

```
class ReadKeys2 {
   public static void main (String args[]) {
      byte buf[] = new byte[80];
      try {
         System.in.read(buf);
      }
      catch (Exception e) {
         System.out.println("Error: " + e.toString());
      }
      String s = new String(buf, 0);
      System.out.println(s);
   }
}
```

/* ReadKeys3.java */

```
class ReadKeys3 {
   public static void main (String args[]) {
      byte buf[] = new byte[10];
      try {
         System.in.read(buf, 0, 10);
      }
      catch (Exception e) {
         System.out.println("Error: " + e.toString());
      }
      String s = new String(buf, 0);
      System.out.println(s);
   }
}
```

THE BUFFEREDINPUTSTREAM CLASS

```
/* ReadKeys4.java */
import java.io.*;
class ReadKeys4 {
   public static void main (String args[]) {
      BufferedInputStream in = new BufferedInputStream(System.in);
      byte buf[] = new byte[10];
      try {
         in.read(buf, 0, 10);
      }
      catch (Exception e) {
         System.out.println("Error: " + e.toString());
      }
      String s = new String(buf, 0);
      System.out.println(s);
   }
}
```

THE DATAINPUTSTREAM CLASS

```
/* ReadFloat.java */
import java.io.*;
class ReadFloat {
   public static void main (String args[]) {
      DataInputStream in = new DataInputStream(System.in);
      String s = new String();
      try {
         s = in.readLine();
         float f = Float.valueOf(s).floatValue();
         System.out.println(f);
      }
      catch (Exception e) {
         System.out.println("Error: " + e.toString());
      }
   }
}
```

THE FILEINPUTSTREAM CLASS

```
/* ReadFile.java */
import java.io.*;
class ReadFile {
   public static void main (String args[]) {
      byte buf[] = new byte[64];
      try {
         FileInputStream in = new FileInputStream("Grocery.txt");
         in.read(buf, 0, 64);
      }
      catch (Exception e) { System.out.println("Error: " + e.toString());    }
      String s = new String(buf, 0);
      System.out.println(s);
   }
}
```

THE STRINGBUFFERINPUTSTREAM CLASS

```
/* ReadString.java */
import java.io.*;
class ReadString {
   public static void main (String args[]) {
      // Get a string of input from the user
      byte buf1[] = new byte[64];
      try {
```

```
            System.in.read(buf1, 0, 64);
        }
        catch (Exception e) {
            System.out.println("Error: " + e.toString());
        }
        String s1 = new String(buf1, 0);
        // Read the string as a string buffer and output it
        StringBufferInputStream in = new StringBufferInputStream(s1);
        byte buf2[] = new byte[64];
        try {
            in.read(buf2, 0, 64);
        }
        catch (Exception e) {
            System.out.println("Error: " + e.toString());
        }
        String s2 = new String(buf2, 0);
        System.out.println(s2);
    }
}
```

OUTPUT STREAM CLASSES
THE OUTPUTSTREAM CLASS

The OutputStream class implements the following methods:

- abstract void write(int b)
- void write(byte b[])
- void write(byte b[], int off, int len)
- void flush()
- void close()

THE PRINTSTREAM CLASS

The PrintStream class also implements a rich set of methods, which follow:

- boolean checkError()
- void print(Object obj)
- synchronized void print(String s)
- synchronized void print(char s[])
- void print(char c)
- void print(int i)
- void print(long l)
- void print(float f)
- void print(double d)
- void print(boolean b)
- void println()
- synchronized void println(Object obj)
- synchronized void println(String s)
- synchronized void println(char s[])
- synchronized void println(char c)
- synchronized void println(int I)
- synchronized void println(long I)
- synchronized void println(float f)
- synchronized void println(double d)
- synchronized void println(boolean b)

THE SYSTEM.OUT OBJECT
THE BUFFEREDOUTPUTSTREAM CLASS

```
/* WriteStuff.java */
import java.io.*;
class WriteStuff {
   public static void main (String args[]) {
      // Copy the string into a byte array
      String s = new String("Dance, spider!\n");
      byte[] buf = new byte[64];
      s.getBytes(0, s.length(), buf, 0);
      // Output the byte array (buffered)
      BufferedOutputStream out = new BufferedOutputStream(System.out);
      try {
         out.write(buf, 0, 64);
         out.flush();
      }
      catch (Exception e) {
         System.out.println("Error: " + e.toString());
      }
   }
}
```

THE DATAOUTPUTSTREAM CLASS

The DataOutputStream class implements the following useful methods beyond those inherited from OutputStream:

- final int size()
- final void writeBoolean(boolean v)
- final void writeByte(int v)
- final void writeShort(int v)
- final void writeChar(int v)
- final void writeInt(int v)
- final void writeLong(long v)
- final void writeFloat(float v)
- final void writeDouble(double v)
- final void writeBytes(String s)
- final void writeChars(String s)

File Classes

THE FILE CLASS

The File class can be instantiated using one of three constructors, which follow:

- File(String path)
- File(String path, String name)
- File(File dir, String name)

The most important methods implemented by the File class follow:

- String getName()
- String getPath()
- String getAbsolutePath()
- String getParent()
- boolean exists()
- boolean canWrite()
- boolean canRead()
- boolean isFile()
- boolean isDirectory()
- boolean isAbsolute()
- long lastModified()
- long length()
- boolean mkdir()
- boolean mkdirs()
- boolean renameTo(File dest)
- boolean delete()
- String[] list()
- String[] list(FilenameFilter filter)

```
/* FileInfo.java */
import java.io.*;
class FileInfo {
   public static void main (String args[]) {
      System.out.println("Enter file name: ");
      char c;
      StringBuffer buf = new StringBuffer();
      try {
         while ((c = (char)System.in.read()) != '\n')
            buf.append(c);
      }
      catch (Exception e) {
         System.out.println("Error: " + e.toString());
      }
      File file = new File(buf.toString());
      if (file.exists()) {
         System.out.println("File Name  : " + file.getName());
         System.out.println("     Path  : " + file.getPath());
         System.out.println("Abs. Path  : " + file.getAbsolutePath());
         System.out.println("Writable   : " + file.canWrite());
         System.out.println("Readable   : " + file.canRead());
         System.out.println("Length     : " + (file.length() / 1024) + "KB");
      }
      else
         System.out.println("Sorry, file not found.");
   }
}
```

THE RANDOMACCESSFILE CLASS

```
/* FilePrint.java */
import java.io.*;
class FilePrint {
   public static void main (String args[]) {
      System.out.println("Enter file name: ");
      char c;
      StringBuffer buf = new StringBuffer();
      try {
         while ((c = (char)System.in.read()) != '\n')
            buf.append(c);
         RandomAccessFile file = new RandomAccessFile(buf.toString(), "rw");
         while (file.getFilePointer() < file.length())
            System.out.println(file.readLine());
      }
      catch (Exception e) {
         System.out.println("Error: " + e.toString());
      }
   }
}
```