

Revision Exercise 2

(1) Fill in the blanks in each of the following.

- a. Class _____ provides methods for drawing.
- b. Java applets begin execution with a series of three method calls: _____ , _____ , and _____ .
- c. Methods _____ and _____ display lines and rectangles.
- d. Keyword _____ is used to indicate that a new class is a subclass of an existing class.
- e. Every Java applet should extend either class _____ or class _____ .
- f. A class definition describes the _____ and _____ of an object.
- g. Java's eight primitive data types are _____ , _____ , _____ , _____ , _____ , _____ , and _____ .

(2) State whether each of the following is *true* or *false*. If *false*, explain why.

- a. Method **drawRect** requires four arguments that specify two points on the applet to draw a rectangle.
- b. Method **drawLine** requires four arguments that specify two points on the applet to draw a line.
- c. Type **Double** is a primitive data type.
- d. Data type **int** is used to declare a floating-point number.
- e. Method **Double.parseDouble** converts a **String** to a **double** floating-point value.

(3) Fill in the blanks in each of the following:

- a. Data type _____ declares a single-precision floating-point variable.
- b. If class **Double** provides method **parseDouble** to convert a **String** to a **double** and class **Integer** provides method **parseInt** to convert a **String** to an **int**, then class **Float** probably provides method _____ to convert a **String** to a **float**.
- c. Data type _____ declares a double-precision floating-point variable.
- d. The _____ or a browser can be used to execute a Java applet.
- e. To load an applet into a browser you must first define an _____ file.
- f. The _____ and _____ tags specify that an applet should be loaded and executed.

(4)

- a. All programs can be written in terms of three types of control structures:
_____, _____ and _____ .
- b. The _____ selection structure is used to execute one action when a condition is true and another action when that condition is false.
- c. Repetition of a set of instructions a specific number of times is called _____ repetition.
- d. When it is not known in advance how many times a set of statements will be repeated, a _____ value can be used to terminate the repetition.

(5) Identify and correct the errors in each of the following:

- a.

```
while ( c <= 5 ) {  
    product *= c;  
    ++c;  
}
```
- b.

```
if ( gender == 1 )  
    System.out.println( "Woman" );  
else;  
    System.out.println( "Man" );
```

(6) What is wrong with the following while repetition structure?

```
while ( z >= 0 )  
  
    sum += z;
```

(7) Identify and correct the errors in each of the following. [Note: There may be more than one error in each piece of code.]

- a.

```
if ( age >= 65 );  
  
    System.out.println( "Age greater than or equal to  
65" );  
  
else  
  
    System.out.println( "Age is less than 65 );
```

```
b. int x = 1, total;
   while ( x <= 10 ) {

       total += x;
       ++x;

   }
```

```
c. While ( x <= 100 )

    total += x;
    ++x;
```

```
d. while ( y > 0 ) {

    System.out.println( y );
    ++y;
```

(8) Write an applet that asks the user to enter two floating-point numbers, obtains the numbers from the user and displays the larger number followed by the words "**is larger**" as a string on the applet. If the numbers are equal, print the message "**These numbers are equal.**"

(9) Write an applet that inputs from the user the radius of a circle as a floating-point number and draws the circle's diameter, circumference and area. Use the constant value 3.14159 for π .

(10) Write a Java application that utilizes looping to print the following table of values:

N	10*N	100*N	1000*N
1	10	100	1000
2	20	200	2000
3	30	300	3000
4	40	400	4000
5	50	500	5000

(11) Drivers are concerned with the mileage obtained by their automobiles. One driver has kept track of several tankfuls of gasoline by recording miles driven and gallons used for each tankful. Develop a Java applet that will input the miles driven and gallons used (both as integers) for each tankful. The program should calculate and display the miles per gallon obtained for each tankful and print the combined miles per gallon obtained for all tankfuls up to this point. All average calculations should produce floating-point results. Using two **TextFields** to input the data..

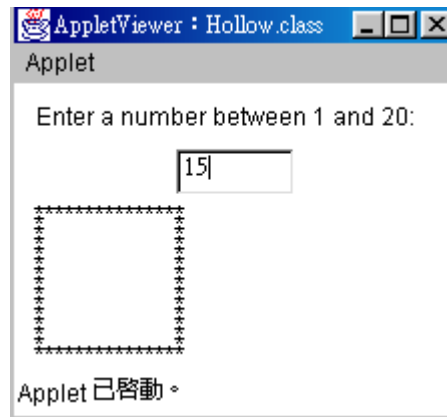


(12) A large company pays its salespeople on a commission basis. The salespeople receive \$200 per week plus 9% of their gross sales for that week. For example, a salesperson who sells \$5000 worth of merchandise in a week receives \$200 plus 9% of \$5000, or a total of \$650. You have been supplied with a list of items sold by each salesperson. The values of these items are as follows:

Item	Value
1	239.99
2	129.75
3	99.95
4	350.89

Develop a Java application that inputs one salesperson's items sold for last week and calculates and displays that salesperson's earnings. There is no limit to the number of items sold by a salesperson.

(13) Write an applet that reads in the size of the side of a square and displays a hollow square of that size out of asterisks using the **drawString** method inside your applet's **paint** method. Use a **TextField** to read the size from the user. Your program should work for squares of all side sizes between 1 and 20. For example , if your program reads a size of 5 , the output should look as follows:



(14) Write an application that keeps displaying in the command window the multiples of the integer 2, namely 2, 4, 8, 16, 32, 64, etc. Your loop should not terminate (i.e., you should create an infinite loop). What happens when you run this program?

(15) Write an application that reads three nonzero integers and determines and prints if they could be the sides of a right triangle.

Answers:

(1) a) **Graphics**. b) **init**, **start** and **paint**. c) **drawLine** and **drawRect**.
d) **extends**. e) **JApplet**, **Applet**. f) attributes and behaviors. g) **byte**, **short**, **int**, **long**, **float**, **double**, **char** and **boolean**.

(2)

- a. False. Method **drawRect** requires four arguments—two that specify the upper-left corner of the rectangle and two that specify the width and height of the rectangle.
- b. True.
- c. False. Type **Double** is a class in the **java.lang** package. Remember that names that start with a capital first letter are normally class names.
- d. False. Data type **double** or data type **float** can be used to declare a floating-point number. Data type **int** is used to declare integers.
- e. True.

(3)

- a. Data type declares a single-precision floating-point variable.
ANS: float.
- b. If class **Double** provides method **parseDouble** to convert a **String** to a **double** and class **Integer** provides method **parseInt** to convert a

String to an **int**, then class **Float** probably provides method to convert a **String** to a **float**.

ANS: parseFloat.

- c. Data type declares a double-precision floating-point variable.

ANS: double.

- b. The or a browser can be used to execute a Java applet.

ANS: appletviewer.

- c. To load an applet into a browser you must first define an file.

ANS: HTML file.

- d. The and tags specify that an applet should be loaded and executed.

ANS: <applet>, </applet>.

(4) a) Sequence, selection and repetition. b) **if/else**. c) Counter-controlled or definite. d) Sentinel, signal, flag or dummy.

(5)

- a. Error: Missing the closing right brace of the **while** body.

Correction: Add closing right brace after the statement **++c;**

- b. Error: Semicolon after **else** results in a logic error. The second output statement will always be executed.

Correction: Remove the semicolon after **else**.

(6) The value of the variable **z** is never changed in the **while** structure. Therefore, if the loop- continuation condition (**z >= 0**) is true, an infinite loop is created. To prevent the infinite loop, **z** must be decremented so that it eventually becomes less than 0.

(7) (a) **ANS:** Semicolon at the end of the **if** condition should be removed. The closing double quote of the second **System.out.println** should be inside of the closing parenthesis.

(b) **ANS:** The variable **total** should be initialized to zero.

(c) **ANS:** The **W** in **While** should be lowercase. The two statements should be enclosed in curly braces to properly group them into the body of the **while**; otherwise the loop will be an infinite loop.

(d) **ANS:** The **++** operator should be changed to **--**. The closing curly brace for the

while loop is missing.

(8)

```
// Larger.java
// Program determines the larger of two numbers
import java.applet.Applet;
import java.awt.*;
import java.awt.event.*;

public class Larger extends Applet implements ActionListener{
    Label prompt1;           // message that prompts user to input first value
    TextField input1;        // input first value here
    Label prompt2;           // message that prompts user to input second value
    TextField input2;        // input second value here
    int number1 , number2;   // store input values

    public void init()
    {
        prompt1 = new Label( "Enter first integer:" );
        add(prompt1);
        // read first number from user as a string
        input1 = new TextField(10);
        add(input1);
        prompt2 = new Label( "Enter second integer and press Enter:" );
        add(prompt2);
        // read second number from user as a string
        input2 = new TextField(10);
        input2.addActionListener(this);
        add(input2);
    }
    //display the results
    public void paint(Graphics g)
    {
        if ( number1 > number2 )
            g.drawString(number1 + " is larger." , 100 , 90);
        else if ( number1 < number2 )
            g.drawString( number2 + " is larger." , 100 , 105);
        else
            g.drawString("These numbers are equal." , 100 , 120);
    }
}
```

```

    }
    // process user's action on the input2 text field
    public void actionPerformed(ActionEvent e)
    {
        // convert numbers from type String to type int
        number1 = Integer.parseInt( input1.getText() );
        number2 = Integer.parseInt( input2.getText() );
        repaint();
    }
}

```

(9)

```

// Circle.java
// Program calculate the area, circumference, and diameter for a circle
import java.applet.Applet;
import java.awt.*;
import java.awt.event.*;

public class Circle extends Applet implements ActionListener{
    int radius;           // radius of circle
    TextField input;      // string entered by user
    String line1,         // strings for output
           line2,
           line3;
    public void init()
    {
        Label prompt;
        prompt = new Label("Enter radius: ");
        add(prompt);
        // read from user as a string
        input = new TextField(10);
        input.addActionListener(this);
        add(input);
    }
    public void actionPerformed(ActionEvent e) {
        // convert number from type String to type int
        radius = Integer.parseInt( input.getText() );
        line1 = "Diameter is " + Math.round( 2 * radius * 100) / 100.00;
        line2 = "Area is " + Math.round( Math.PI * radius * radius * 100) / 100.00;
    }
}

```



```

        line3 = "Circumference is " + Math.round( 2 * Math.PI * radius * 100) / 100.00;
        repaint();
    }
    public void paint( Graphics g )
    {
        g.drawString( line1, 25, 100 );
        g.drawString( line2, 25, 120 );
        g.drawString( line3, 25, 140 );
    }
}

```

(10)

```

// Table.java
// Program prints a table of values
public class Table {
    public static void main( String args[] )
    {
        int n = 0;
        System.out.println( "N\t10*N\t100*N\t1000*N\n" );
        while ( ++n <= 5 )
            System.out.println( n + "\t" + ( 10 * n ) +
                                "\t" + ( 100 * n ) + "\t"
                                + ( 1000 * n ) );
    }
}

```

(11)

```

// Gas.java
// Program calculates average mpg
import java.applet.Applet;
import java.awt.*;
import java.awt.event.*;
import javabook.*;

public class Gas extends Applet implements ActionListener{
    int miles, gallons;
    int totalMiles = 0, totalGallons = 0;
    float milesPerGallon;
    float totalMilesPerGallon;
}

```

```

        TextField input1, input2;
        String result = "";

public void init()
{
    Label prompt1 ,
        prompt2;
    prompt1 = new Label("Enter Miles: ");
    add(prompt1);
    // read from user as a string
    input1 = new TextField(10);
    add(input1);
    prompt2 = new Label("Enter Gallons: ");
    add(prompt2);
    // read from user as a string
    input2 = new TextField(10);
    input2.addActionListener(this);
    add(input2);
}

public void actionPerformed( ActionEvent e )
{
    // convert numbers from type String to type int
    miles = Integer.parseInt( input1.getText() );
    gallons = Integer.parseInt( input2.getText());
    repaint();
}

public void paint(Graphics g) {
    totalMiles += miles;
    totalGallons += gallons ;
    if ( gallons != 0 ) {
        milesPerGallon = ( float ) miles / gallons;
        result = "MPG this tankful: " + Math.round(milesPerGallon *
1000) / 1000.00 ;
    }
    g.drawString(result , 10 , 90);
    result = "";
    if ( totalGallons != 0 ) { // never will be 0 due to test above

```

```

        totalMilesPerGallon = ( float ) totalMiles / totalGallons;
        result += "Total MPG: " + Math.round(totalMilesPerGallon *
1000) / 1000.00;
    }
    g.drawString(result , 10 , 120);
}
}

```

(12)

// Sales.java

// Program calculates sales

import javabook.*;

public class Sales {

public static void main(String args[])

{

MainWindow mainWindow = new MainWindow("Sales Earnings");

InputBox inputBox = new InputBox(mainWindow);

OutputBox outputBox = new OutputBox(mainWindow);

mainWindow.show();

outputBox.show();

double gross = 0.0, earnings;

int product = 0, number;

while (product < 4) {

product++;

// read number from user as a string

number = inputBox.getInteger("Enter number sold of product #" +
product + ":");

if (product == 1)

gross = gross + number * 239.99;

else if (product == 2)

gross = gross + number * 129.75;

else if (product == 3)

gross = gross + number * 99.95;

else if (product == 4)

gross = gross + number * 350.89;

}

earnings = 0.09 * gross + 200;

String result = "Earnings this week: " + Math.round(earnings * 100) / 100.00;

```

        outputBox.println(result);
    }
}
(13)
// Hollow.java
// Program prints a hollow square
import java.applet.Applet;
import java.awt.*;
import java.awt.event.*;
public class Hollow extends Applet implements ActionListener{
    int stars;
    TextField input;          // for user input
    public void init()
    {
        Label prompt;
        stars = 0;
        prompt = new Label("Enter a number between 1 and 20:");
        add(prompt);
        // read number from user as a string
        input = new TextField( 5);
        input.addActionListener(this);
        add(input);
    }
    public void actionPerformed(ActionEvent e) {
        // convert numbers from type String to type int
        stars = Integer.parseInt( input .getText());
        repaint();
    }
    public void paint( Graphics g )
    {
        if ( stars < 1 ) {
            stars = 1;
            g.drawString("Error!! Using default value 1." , 100 , 40);
        }
        else if ( stars > 20 ) {
            stars = 20;
            g.drawString("Error!! Using default value 20." , 100 , 40);
        }
    }
}

```

```

int x = 5, y = 70;
int i = 1, j = 1;
while ( i <= stars ) {
    while ( j <= stars ) {
        if ( i == 1 )
            g.drawString( "*", x += 5, y );
        else if ( i == stars )
            g.drawString( "*", x += 5, y );
        else if ( j == 1 )
            g.drawString( "*", x += 5, y );
        else if ( j == stars )
            g.drawString( "*", x += 5, y );
        else
            g.drawString( " ", x += 5, y );
        j++;
    }
    j = 1;
    i++;
    y += 5;
    x = 5;
}
}
}

```

(14)

```

// Infinite.java
// Program creates an infinite loop
// Runs out of integers (number is too big)
public class Infinite {
    public static void main( String args[] )
    {
        int x = 1;
        while ( true ) {
            x *= 2;
            System.out.println( x + " " );
        }
    }
}

```

(15)

```

// Triangle.java
// Program takes three integers and
// determines if they form the sides
// of a right triangle
import javabook.*;
public class Triangle {
    public static void main( String args[] )
    {
        MainWindow mainWindow = new MainWindow("Test Triangle  program");
        InputBox  inputBox      = new InputBox(mainWindow);
        OutputBox outputBox      = new OutputBox(mainWindow);
        mainWindow.show();
        outputBox.show();
        int side1, side2, hypotenuse;
        String  result;
        // read number from user as a string
        side1 = inputBox.getInteger( "Enter length of side 1:" );
        // read number from user as a string
        side2 = inputBox.getInteger( "Enter length of side 2:" );
        // read number from user as a string
        hypotenuse =  inputBox.getInteger( "Enter hypotenuse:" );
        if ( ( side1 * side1 + side2 * side2 ) == ( hypotenuse * hypotenuse ) )
            result = "These are the sides of a right triangle.";
        else
            result = "These do not form a right triangle.";
        outputBox.println(result);
    }
}

```