

Lab Goal : This lab was designed to teach you how to instantiate an object, pass parameters, calculate values, and display the results.

Lab Description : Calculate the slope of a line when given 2 points. The formula for the slope of a line is as follows :: $m = \frac{y_2 - y_1}{x_2 - x_1}$

Sample Data:

```
1 9 14 2
1 7 18 3
6 4 2 2
4 4 5 3
1 1 2 9
```

Files Needed ::

```
Line.java
LineRunner.java
```

Sample Output :

```
the slope is -0.54
the slope is -0.24
the slope is 0.50
the slope is -1.00
the slope is 8.00
```

FORMATTING OUTPUT

```
double dec = 9.541724;

out.printf("%.3f\n",dec);           //outs 9.542
    //printf is a void method

out.println(String.format("%.3f",dec)); //outs 9.542
    //format is a String return method
    //format is useful when writing toString() methods
```

Lab Goal : This lab was designed to teach you how to instantiate an object, pass parameters, calculate values, and display the results.

This lab also introduces the idea of a return method. You may have to do some research in order to complete this lab.

Lab Description : Convert Celsius to Fahrenheit.

- Subtract 32° to adjust for the difference in the Fahrenheit scale.
Multiply the result by 5/9.
- Example: convert 98.6° Fahrenheit to Celsius.
 $98.6 - 32 = 66.6$
 $66.6 * 5/9 = 333/9 = 37° C.$

Sample Data:

98.6
52.30
82.45
75.00
100.00

Files Needed ::

Fahrenheit.java
FahrenheitRunner.java

Sample Output :

98.60 degrees Fahrenheit == 37.00 degrees Celsius

52.30 degrees Fahrenheit == 11.28 degrees Celsius

82.45 degrees Fahrenheit == 28.03 degrees Celsius

75.00 degrees Fahrenheit == 23.89 degrees Celsius

100.00 degrees Fahrenheit == 37.78 degrees Celsius

FORMATTING OUTPUT

```
double dec = 9.541724;

out.printf("%.3f\n",dec);           //outs 9.542
    //printf is a void method

out.println(String.format("%.3f",dec)); //outs 9.542
    //format is a String return method
    //format is useful when writing toString() methods
```

Lab Goal : This lab was designed to teach you more about objects, input, output formatting, and the Math class.

Lab Description : Determine the distance the between two points.

The distance formula for (x1, y1) and (x2, y2) is:

$$\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Sample Data :

```
1 1 2 1
1 1 -2 2
1 1 0 0
```

Sample Output :

```
Enter X1 :: 1
Enter Y1 :: 1
Enter X2 :: 2
Enter Y2 :: 1
distance == 1.000
```

```
Enter X1 :: 1
Enter Y1 :: 1
Enter X2 :: -2
Enter Y2 :: 2
distance == 3.162
```

```
Enter X1 :: 1
Enter Y1 :: 1
Enter X2 :: 0
Enter Y2 :: 0
distance == 1.414
```

Files Needed ::

```
Distance.java
DistanceRunner.java
```

FORMATTING OUTPUT

```
                //printf is a void method
System.out.printf("%.3f\n",9.541724);                //outs 9.542

                //format is a String return method
System.out.println(String.format("%.3f",9.541724)); //outs 9.542
```

Lab Goal : This lab was designed to check for understanding of strings and string methods.

Lab Description : Given a String, return a specified section from the String.

HINTS :: substring() will be very useful for this lab.

Sample Data :

```
chicken
3 6
Alligator
3 8
COMPUTER SCIENCE IS THE BEST!
9 12
0 12
5 12
I like fried chicken and mashed potatoes!
0 7
7 15
15 26
22 26
```

Files Needed ::

```
StringRipper.java
RipperRunner.java
```

Sample Output :

```
cke
chicken
```

```
igato
alligator
```

```
SCI
COMPUTER SCI
TER SCIENCE IS
COMPUTER SCIENCE IS THE BEST!
```

```
I like
fried ch
icken and m
nd m
I like fried chicken and mashed potatoes!
```

Lab Goal : This lab was designed to teach you how to use nested loops. You will also learn how to make some cool shapes.

Lab Description : Used nested loops to print out a letter pattern as shown below. Wrap around the letters from Z to A.

Sample Data :

C 4
A 5
B 7
X 6
Z 8

Files Needed ::

TriangleFive.java
TriangleFiveRunner.java

Sample Output :

```
CCCC DDD EE F
CCCC DDD EE
CCCC DDD
CCCC
```

```
AAAAA BBBB CCC DD E
AAAAA BBBB CCC DD
AAAAA BBBB CCC
AAAAA BBBB
AAAAA
```

```
BBBBBBB CCCCCC DDDDD EEEE FFF GG H
BBBBBBB CCCCCC DDDDD EEEE FFF GG
BBBBBBB CCCCCC DDDDD EEEE FFF
BBBBBBB CCCCCC DDDDD EEEE
BBBBBBB CCCCCC DDDDD
BBBBBBB CCCCCC
BBBBBBB
```

```
XXXXXX YYYYY ZZZZ AAA BB C
XXXXXX YYYYY ZZZZ AAA BB
XXXXXX YYYYY ZZZZ AAA
XXXXXX YYYYY ZZZZ
XXXXXX YYYYY
XXXXXX
```

```
ZZZZZZZZ AAAAAA BBBBBB CCCCC DDDD EEE FF G
ZZZZZZZZ AAAAAA BBBBBB CCCCC DDDD EEE FF
ZZZZZZZZ AAAAAA BBBBBB CCCCC DDDD EEE
ZZZZZZZZ AAAAAA BBBBBB CCCCC DDDD
ZZZZZZZZ AAAAAA BBBBBB CCCCC
ZZZZZZZZ AAAAAA BBBBBB
ZZZZZZZZ AAAAAA
ZZZZZZZZ
```

```
// A+ Computer Science - www.apluscompsci.com
//Name -
//Date -
//Class -
//Lab -

import static java.lang.System.*;

public class TriangleFiveRunner
{
    public static void main(String args[])
    {
        TriangleFive test = new TriangleFive('C',4);
        out.println(test);

        test.setLetter('A');
        test.setAmount(5);
        out.println(test);

        test.setLetter('B');
        test.setAmount(7);
        out.println(test);

        test.setLetter('X');
        test.setAmount(6);
        out.println(test);

        test.setLetter('Z');
        test.setAmount(8);
        out.println(test);
    }
}
```

```
//(c) A+ Computer Science
//www.apluscompsci.com

//Name -
//Date -
//Class -
//Lab -

import static java.lang.System.*;

public class RipperRunner
{
    public static void main ( String[] args )
    {
        StringRipper demo = new StringRipper("chicken");
        out.println(demo.ripString(3,6));
        out.println(demo);

        demo.setString("alligator");
        out.println(demo.ripString(3,8));
        out.println(demo);

        demo.setString("COMPUTER SCIENCE IS THE BEST!");
        out.println(demo.ripString(9,12));
        out.println(demo.ripString(0,12));
        out.println(demo.ripString(5,20));
        out.println(demo);

        demo.setString("I like fried chicken and mashed potatoes!");
        out.println(demo.ripString(0,7));
        out.println(demo.ripString(7,15));
        out.println(demo.ripString(15,26));
        out.println(demo.ripString(22,26));
        out.println(demo);
    }
}
```

```
// A+ Computer Science - www.apluscompsci.com
//Name -
//Date -
//Class -
//Lab -
```

```
public class FahrenheitRunner
{
    public static void main( String[] args )
    {
        Fahrenheit fahr = new Fahrenheit();
        fahr.setFahrenheit(98.6);
        fahr.print();

        fahr.setFahrenheit(52.3);
        fahr.print();

        fahr.setFahrenheit(82.45);
        fahr.print();

        fahr.setFahrenheit(75);
        fahr.print();

        fahr.setFahrenheit(100);
        fahr.print();
    }
}
```



```
// A+ Computer Science - www.apluscompsci.com
//Name -
//Date -
//Class -
//Lab -

import java.util.Scanner;
import static java.lang.System.*;
import static java.lang.Math.*;

public class DistanceRunner
{
    public static void main( String[] args )
    {
        Scanner keyboard = new Scanner( System.in );

        out.print("Enter X1 :: ");
        int x1 = keyboard.nextInt();
        out.print("Enter Y1 :: ");
        int y1 = keyboard.nextInt();
        out.print("Enter X2 :: ");
        int x2 = keyboard.nextInt();
        out.print("Enter Y2 :: ");
        int y2 = keyboard.nextInt();

        Distance test = new Distance();
        test.setCoordinates(x1, y1, x2, y2);
        test.calcDistance();
        test.print();

        out.print("Enter X1 :: ");
        x1 = keyboard.nextInt();
        out.print("Enter Y1 :: ");
        y1 = keyboard.nextInt();
        out.print("Enter X2 :: ");
        x2 = keyboard.nextInt();
        out.print("Enter Y2 :: ");
        y2 = keyboard.nextInt();

        test.setCoordinates(x1, y1, x2, y2);
        test.calcDistance();
        test.print();

        out.print("Enter X1 :: ");
        x1 = keyboard.nextInt();
        out.print("Enter Y1 :: ");
        y1 = keyboard.nextInt();
        out.print("Enter X2 :: ");
        x2 = keyboard.nextInt();
        out.print("Enter Y2 :: ");
        y2 = keyboard.nextInt();

        test.setCoordinates(x1, y1, x2, y2);
        test.calcDistance();
        test.print();
    }
}
```

```
// A+ Computer Science - www.apluscompsci.com
//Name -
//Date -
//Class -
//Lab -

public class LineRunner
{
    public static void main( String[] args )
    {
        Line line = new Line(1,9,14,2);
        line.calculateSlope();
        line.print();

        line.setCoordinates(1,7,18,3);
        line.calculateSlope();
        line.print();

        line.setCoordinates(6,4,2,2);
        line.calculateSlope();
        line.print();

        line.setCoordinates(4,4,5,3);
        line.calculateSlope();
        line.print();

        line.setCoordinates(1,1,2,9);
        line.calculateSlope();
        line.print();
    }
}
```

Name: _____

Summer, 2018

AP Java Summer Assignment A.**AP Java Multiple Choice Questions:**

You may write on each packet (chapter). Show evidence / give explanation of why you chose that answer. **No work, no credit!**

Questions	*On Time	Possible Points	Points Earned
Chapter 1		24	
Chapter 2		24	
Chapter 3		24	
Chapter 4		24	
Chapter 6		24	
Total Points		120	

BlueJ - Labs

Lab	Methods	*On Time	Possible Points	Points Earned
Line	Line() (constructor - 4 parameters)		2	
	setCoordinates() (4 parameters)		2	
	calculateSlope()		2	
	print()		2	
Fahrenheit	Fahrenheit() (constructor - default)		2	
	setFahrenheit() (1 parameter)		2	
	getCelsius()		2	
	print()		2	
Distance	Distance() (constructor - default)		2	
	Distance() (constructor - 4 parameters)		2	
	setCoordinates() (4 parameters)		2	
	calcDistance()		2	
	print()		2	
Ripper	StringRipper() (constructor - default)		2	
	StringRipper() (constructor - 1 parameter)		2	
	setString() (1 parameter)		2	
	ripString() (2 parameters)		2	
	toString()		2	

Name	Name() (constructor - default)		2	
	Name() (constructor - 1 parameter)		2	
	setName() (1 parameter)		2	
	getFirst()		2	
	getLast()		2	
	toString()		2	
TriangleFive	TriangleFive() (constructor - default)		2	
	TriangleFive() (constructor - 2 parameters)		2	
	setLetter() (1 parameter)		2	
	setAmount() (1 parameter)		2	
	toString()		4	
Total Points			60	

* 10 point deduction for each part if not handed in on the first day of class.

Chapter 1 Questions Control Structures

01. Consider the following code segment.

```
int p = 0;
while (p < 5)
{
    int q = 0;
    while (q < 4)
    {
        p += q;
        q++;
        System.out.println(p + " " + q);
    }
}
```

What is the last output when the segment executes?

- (A) 0 1
- (B) 1 2
- (C) 3 3
- (D) 6 4
- (E) 10 5

02. Consider the following code segment.

```
int count = 5;
for (int p = 0; p < 7; p++)
{
    if (p % 3 == 0)
        count--;
    else
        count ++;
}
System.out.println(count);
```

What is printed as a result of executing the code segment?

- (A) 4
- (B) 5
- (C) 6
- (D) 7
- (E) 8

03. Consider the following code segment.

```
int x = <some integer greater than zero>
int n = 100;

if (x < 1000)
{
    if (x > 1000)
        n = 200;
    else
        n = 300;
}
else
{
    if (x < 1000)
        n = 400;
    else
        n = 300;
}
System.out.println(n);
```

What is printed as a result of executing the code segment?

- (A) 100
- (B) 200
- (C) 300
- (D) 400
- (E) 200 or 400

04. Consider the following code segment.

```
int x = 0;
for (int y = 1; y <= 12; y++)
{
    while (x <= y)
    {
        if (x % 2 == 0)
            x += 2;
        else
            y -= 2;
    }
}
System.out.println(x);
```

What is printed as a result of executing the code segment?

- (A) 11
- (B) 12
- (C) 13
- (D) 14
- (E) 15

05. Consider the following code segment.

```
int q = 0;
for (int p = 0; p < 5; p++)
{
    q = q + p;
}
System.out.println(q);
```

What is printed as a result of executing the code segment?

- (A) 1
- (B) 3
- (C) 6
- (D) 10
- (E) 15

06. Consider the following code segment.

```
int q = 0;
for (int p = 0; p < 10; p++)
{
    q++;
    p += q;
    System.out.print(p + " ");
}
```

What is printed as a result of executing the code segment?

- (A) 1 4 8 13
- (B) 1 4 8 13 19
- (C) 1 3 5 7 9
- (D) 1 3 5 7 9 11
- (E) 0 2 4 6 8 10

07. Consider the following code segment.

```
int q = <some integer value greater than 0 >
int p = <some integer value greater than q >
while (p > q)
{
    if (p % 2 == 0)
        p--;
    else
        q++;
}
System.out.println(q + " " + p);
```

What kind of values are printed when the segment executes?

- (A) Two positive integers, such that p equals q
- (B) Two positive integers, which are the initial values of p and q
- (C) Two positive integers, such that p is greater than q
- (D) Two positive integers, such that p is lesser than q
- (E) Two positive integers, such that p equals q + 1

08. Consider the following code segment.

```
int k = 0;
for (k = 1; k <= 5; k++)
{
    k++;
}
k++;
System.out.println(k);
```

What is printed as a result of executing the code segment?

- (A) 5
- (B) 6
- (C) 7
- (D) 8
- (E) 9

09. Consider the following code segment.

```
int count = 0;
for (int p = 0; p <= 5; p++)
{
    for (int q = p; q <= 5; q++)
    {
        count++;
    }
}
System.out.println(count);
```

What is printed as a result of executing the code segment?

- (A) 3
- (B) 6
- (C) 10
- (D) 15
- (E) 21

10. Consider the following code segment.

```
int n = 6;
int count = 0;
for (int p = 1; p <= n-1; p++)
    for (int q = 0; q < p; q++)
        count++;
System.out.println(count);
```

What is the value of count when the code segment finishes executing?

- (A) 10
- (B) 15
- (C) 21
- (D) 28
- (E) 36

11. Consider the following code segment.

```
int p = 10;
int q = 5;
while (p > 0 && q > 0)
{
    p++;
    if (p % 2 == 0)
        q--;
    System.out.println(p + " " + q);
}
```

What is the last output when the segment executes?

- (A) 16 2
 - (B) 17 2
 - (C) 18 1
 - (D) 19 1
 - (E) 20 0
12. Consider the following code segment.

```
int p = 10;
int q = 5;
while (q != 0 && p/q > 0)
{
    p--;
    System.out.println(p + " " + q);
    q--;
}
```

What is the last output when the segment executes?

- (A) 7 3
- (B) 6 2
- (C) 5 1
- (D) 4 0
- (E) An ArithmeticException error message due to division by zero

Chapter 2 Questions Methods and Parameters

01. Consider the following method.

```
/** Precondition: p > 0
 */
public static int method0201(int p)
{
    int count = 1;
    for (int q = 1; q < p; q++)
    {
        count += count;
    }
    return count;
}
```

What value is returned as a result of the call `method0201(n)` ?

- (A) n
- (B) $2n$
- (C) n^2
- (D) 2^n
- (E) $2^{(n-1)}$

02. Consider the following method.

```
/** Precondition: p > 0
 */
public static int method0202(int p)
{
    int count = 1;
    for (int q = 1; q < p; q++)
        count += count;
    return count;
}
```

What value is returned as a result of the call `method0202(5)` ?

- (A) 5
- (B) 7
- (C) 8
- (D) 15
- (E) 16

03. Consider the following code segment and method.

```
for (int n = 1; n <= 10; n++)
    System.out.print(method0203(n) + " ");

public static int method0203(int n)
{
    int temp1 = 0;
    int temp2 = 1;
    int temp3 = 1;
    for (int k = 3; k <= n; k++)
    {
        temp3 = temp1 + temp2;
        temp1 = temp2;
        temp2 = temp3;
    }
    return temp3;
}
```

What is printed as a result of executing the code segment?

- (A) 55 34 13 8 5 3 2 1 1
- (B) 1 1 2 3 5 8 13 34 55
- (C) 1 1 1 2 3 5 8 13 21 34
- (D) 34 21 13 8 5 3 2 1 1 1
- (E) 0 1 1 2 3 5 8 13 21 34

04. Consider the following method.

```
public static int method0204(int n)
{
    int k1 = 2;
    int k2 = 3;
    int k3 = 4;
    for (int p = 1; p <= n; p++)
    {
        k1 = k2;
        k2 = k3;
        k3 = k1 + k2;
    }
    return k3;
}
```

What value is returned as a result of the call method0204(5)?

- (A) 23
- (B) 35
- (C) 47
- (D) 62
- (E) 71

05. Consider the following code segment and method.

```
int x = 5;
x = method0205(x);
System.out.println(x);

public static int method0205(int n)
{
    for (int k = n; k <= 10; k++)
        n += k;
    return n;
}
```

What is printed as a result of executing the code segment?

- (A) 50
- (B) 40
- (C) 31
- (D) 23
- (E) 16

06. Consider the following code segment and method.

```
int x = 10;
int y = 20;
swap(x, y);
System.out.println(x + " " + y);

public static void swap(int p, int q)
{
    int t = p;
    p = q;
    q = t;
}
```

What is printed as a result of executing the code segment?

- (A) 20 10
- (B) 10 20
- (C) 10 10
- (D) 20 20
- (E) 0 0

07. Consider the following two methods.

```
/** Precondition: n1 > 0
 *                n2 > 0
 */
public static int method0207a(int n1, int n2)
{
    int temp = method0207b(n1,n2);
    return n1 / temp * n2;
}
```

```
/** Precondition: p > 0
 *                q > 0
 */
public static int method0207b(int p, int q)
{
    int rem = 1;
    int k = 0;
    while (rem != 0)
    {
        rem = p % q;
        if (rem == 0)
        {
            k = q;
        }
        else
        {
            p = q;
            q = rem;
        }
    }
    return k;
}
```

What value is returned as a result of the call `method0207a(30, 45)` ?

- (A) 30
- (B) 45
- (C) 90
- (D) 150
- (E) 450

08. Consider the following code segment and class.

```
Widget w1 = new Widget(66);  
Widget w2 = new Widget(77);  
System.out.println(w2.getWidgets() + " " + w1.getWidgets());  
System.out.println(w2 + " " + w1);
```

```
class Widget  
{  
    private int numWidgets;  
  
    public Widget(int nW)  
    {  
        numWidgets = nW;  
    }  
  
    public int getWidgets()  
    {  
        return numWidgets;  
    }  
}
```

What is printed as a result of executing the code segment?

Note: Memory reference values will fluctuate with different executions.

- (A) 66 77
Widget@2e816 Widget@addbf1
- (B) 77 66
77 66
- (C) 77 66
Widget@addbf1 Widget@42e816
- (D) 66 77
66 77
- (E) 0 0
0 0

09. Consider the following code segment and class.

```
Widget w1 = new Widget(66);
Widget w2 = new Widget(77);
System.out.println(w1.getWidgets() + " " + w2.getWidgets());
System.out.println(w1 + " " + w2);
```

```
class Widget
{
    private int numWidgets;
    public Widget(int numWidgets)    { numWidgets = numWidgets; }
    public int getWidgets()          { return numWidgets; }
}
```

What is printed as a result of executing the code segment?

Note: Memory reference values will fluctuate with different executions.

- (A) 66 77
Widget@42e816 Widget@adbf1
- (B) 66 77
66 77
- (C) 77 66
Widget@adbf1 Widget@42e816
- (D) 0 0
Widget@adbf1 Widget@42e816
- (E) 0 0
0 0

10. Consider the following program.

```
public class DS0210
{
    public static void main(String args[]) { samba(65.0); }
    public static void samba(int k)       { System.out.println(k); }
    public static void samba(double k)    { System.out.println(k); }
    public static void samba(char k)      { System.out.println(k); }
    public static void samba(String k)    { System.out.println(k); }
}
```

What is printed as a result of executing the program?

- (A) 65
- (B) 65.0
- (C) k
- (D) A
- (E) A Duplicate method compile error message

11. Consider the following method.

```
public static int method0211 (int n)
{
    int temp = 1;
    for (int k = n; k > 1; k--)
        temp *= k;
    return temp;
}
```

What value is returned as a result of the call `method0211 (5)` ?

- (A) 2
- (B) 6
- (C) 24
- (D) 120
- (E) 720

12. Consider the following program.

```
public class DS0212
{
    public static void main(String args[])
    {
        waltz("Hello",100,Math.PI);
    }
    public static void waltz(String n)
    {
        System.out.println(n);
    }
    public static void waltz(String p, int q)
    {
        System.out.println(p + " " + q);
    }
    public static void waltz(String x, int y, double z)
    {
        System.out.println(x + " " + y + " " + z);
    }
    public static void waltz(String a, int b, double c, int d)
    {
        System.out.println(a + " " + b + " " + c + " " + d);
    }
}
```

What is printed as a result of executing the program?

- (A) Hello
- (B) Hello 100
- (C) Hello 100 3.141592653589793
- (D) An Unknown symbol compile error message
- (E) A Duplicate method compile error message



Chapter 3 Questions Boolean Algebra

01. The Boolean expression

$(A \ || \ B) \ \&\& \ A$

is true

- (A) in all cases.
- (B) whenever A is true.
- (C) whenever B is true.
- (D) whenever either A is true or B is true.
- (E) whenever both A is true and B is true.

02. The Boolean expression

$(A \ \&\& \ B) \ || \ B$

is true

- (A) in all cases.
- (B) whenever A is true.
- (C) whenever B is true.
- (D) whenever either A is true or B is true.
- (E) whenever both A is true and B is true.

03. The Boolean expression

$(A \ \&\& \ B) \ || \ (A \ \&\& \ B)$

is true

- (A) in all cases.
- (B) whenever A is true.
- (C) whenever B is true.
- (D) whenever either A is true or B is true.
- (E) whenever both A is true and B is true.

04. The Boolean expression

$(A \vee B) \vee A$

is

- (A) false whenever A is true.
- (B) true whenever B is false.
- (C) true whenever either A is true or B is true.
- (D) true whenever both A is false and B is false.
- (E) never true.

05. Which Boolean law is demonstrated by

$\neg(A \vee B)$ is equivalent to $\neg A \wedge \neg B$

- (A) Distributive property
- (B) Commutative property
- (C) DeMorgan's Law
- (D) Descartes' Law
- (E) Euclid's Law

06. Which Boolean law is demonstrated by

$\neg(A \wedge B)$ is equivalent to $\neg A \vee \neg B$

- (A) Distributive property
- (B) Commutative property
- (C) DeMorgan's Law
- (D) Descartes' Law
- (E) Euclid's Law

07. The Boolean expression

$(A \ || \ B) \ || \ !(A \ || \ B)$

evaluates to

- (A) false in all cases.
- (B) true whenever only A is true or only B is true.
- (C) true whenever A is true or B is true.
- (D) true whenever both A is true and B is true.
- (E) true in all cases.

08. The Boolean expression

$!(A \ || \ B)$

evaluates to

- (A) false in all cases.
- (B) true in all cases.
- (C) true whenever only A is true or only B is true.
- (D) true whenever A is true or B is true.
- (E) true whenever both A is false and B is false.

09. The Boolean expression

$!((A < B) \ || \ (C > D))$

is equivalent to which of the following expressions?

- (A) $(A < B) \ \&\& \ (C > D)$
- (B) $(A \geq B) \ || \ (C \leq D)$
- (C) $(A > B) \ \&\& \ (C < D)$
- (D) $(A > B) \ || \ (C < D)$
- (E) $(A \geq B) \ \&\& \ (C \leq D)$

10. The Boolean expression

`(A || B) && (!A && !B)`

evaluates to

- (A) false in all cases.
- (B) true in all cases.
- (C) true whenever only A is true or only B is true.
- (D) true whenever A is true or B is true.
- (E) true whenever both A is false and B is false.

11. Consider the following program segment.

```
int p = <some integer greater than zero>
int q = <some integer greater than zero>
while (p != 0 || q != 0)
{
    p--;
    q--;
    niftyMethod();
}
```

For which values of p and q will the loop structure condition short-circuit?

- (A) Whenever p does not equal 0
- (B) Whenever q does not equal 0
- (C) Only whenever both p equals 0 and q equals 0
- (D) Only whenever p equals 0 or q equals 0
- (E) The structure will never short-circuit.

12. The Boolean expression

`(A || B) && (!A || !B)`

evaluates to

- (A) false in all cases.
- (B) true in all cases.
- (C) true whenever only A is true or only B is true.
- (D) true whenever A is true or B is true.
- (E) true whenever both A is true and B is true.

Chapter 4 Questions

Focus on OOP-1, Encapsulation

01. Consider the following code segment and class.

```
Student kathy = new Student();
System.out.println(kathy.getAge());

class Student
{
    private int age;
    public int getAge() { return age; }
}
```

What is printed as a result of executing the code segment?

- (A) a random number in the int range
 - (B) a memory reference value
 - (C) 0
 - (D) a compile error message
 - (E) an Exception error message
02. Consider the following code segment and class.

```
Student kathy = new Student(10);
System.out.println(kathy.getAge());

class Student
{
    private int age;
    public Student(int a) { age = a; }
    public int getAge() { return age; }
}
```

What is printed as a result of executing the code segment?

- (A) 10
- (B) a memory reference value
- (C) 0
- (D) a compile error message
- (E) an Exception error message

03. Consider the following code segment and class.

```
Student kathy = new Student(10);  
System.out.println(kathy.getAge());
```

```
class Student  
{  
    private int age;  
    public Student(int age) { age = age; }  
    public int getAge()     { return age; }  
}
```

What is printed as a result of executing the code segment?

- (A) 10
- (B) a memory reference value
- (C) 0
- (D) a compile error message
- (E) an Exception error message

04. Consider the following code segment and class.

```
Student kathy = new Student(10);  
System.out.println(kathy.getAge());
```

```
class Student  
{  
    private int age;  
    public Student(int age) { this.age = age; }  
    public int getAge()     { return age; }  
}
```

What is printed as a result of executing the code segment?

- (A) 10
- (B) a memory reference value
- (C) 0
- (D) a compile error message
- (E) an Exception error message

05. Consider the following code segment and class.

```
Student kathy = new Student(10);
System.out.println(kathy.getAge());
```

```
class Student
{
    private int age;

    public Student(int age)
    {
        this.age = age;
    }

    public int getAge()
    {
        int age;
        return age;
    }
}
```

What is printed as a result of executing the code segment?

- (A) 10
- (B) a memory reference value
- (C) 0
- (D) a compile error message
- (E) an Exception error message

06. Consider the following code segment and class.

```
Student kathy = new Student();
System.out.println(kathy.age);
```

```
class Student
{
    int age;

    public Student()
    {
        age = 10;
    }
}
```

What is wrong with this program?

- (A) It does not compile, because `age` cannot be accessed directly.
- (B) The intent of encapsulation is violated, because `age` can be accessed directly.
- (C) The constructor has no parameter.
- (D) There is no value assigned to the `age` instance variable.
- (E) There is nothing wrong with the program.

07. Consider the following code segment and class.

```
Bank tom = new Bank(500.0);  
tom.makeDeposit(300.0);  
tom.makeDeposit(175.0);  
tom.makeWithdrawal(1000.0);  
tom.makeDeposit(100.0);  
System.out.println(tom.getBalance());
```

```
class Bank  
{  
    private double balance;  
  
    public Bank(double amount)  
    {  
        balance = amount;  
    }  
  
    public double getBalance()  
    {  
        return balance;  
    }  
  
    public void makeDeposit(double amount)  
    {  
        balance += amount;  
    }  
  
    public void makeWithdrawal(double amount)  
    {  
        if (amount > balance)  
            balance -= 35.0;  
        else  
            balance -= amount;  
    }  
}
```

What is printed as a result of executing the code segment?

- (A) 40.0
- (B) 75.0
- (C) 1040.0
- (D) -40.0
- (E) -75.0

08. Consider the following code segment and class.

```
Bank tom = new Bank(500.0);  
Bank sue = new Bank(500.0);  
sue.makeDeposit(225.0);  
tom.makeWithdrawal(100.0);  
System.out.println(tom.getBalance() + " " + sue.getBalance());
```

```
class Bank  
{  
    private double balance;  
  
    public Bank(double amount)  
    {  
        this.balance = amount;  
    }  
  
    public double getBalance()  
    {  
        return this.balance;  
    }  
  
    public void makeDeposit(double amount)  
    {  
        this.balance += amount;  
    }  
  
    public void makeWithdrawal(double amount)  
    {  
        this.balance -= amount;  
    }  
}
```

What is printed as a result of executing the code segment?

- (A) 625.0 625.0
- (B) 400.0 775.0
- (C) 725.0 400.0
- (D) 400.0 400.0
- (E) 725.0 725.0

09. Consider the following code segment and class.

```
Bank tom = new Bank(500.0);  
Bank sue = tom;  
sue.makeDeposit(225.0);  
tom.makeWithdrawal(100.0);  
System.out.println(tom.getBalance() + " " + sue.getBalance());
```

```
class Bank  
{  
    private double balance;  
  
    public Bank(double amount)  
    {  
        balance = amount;  
    }  
  
    public double getBalance()  
    {  
        return balance;  
    }  
  
    public void makeDeposit(double amount)  
    {  
        balance += amount;  
    }  
  
    public void makeWithdrawal(double amount)  
    {  
        balance -= amount;  
    }  
}
```

What is printed as a result of executing the code segment?

- (A) 625.0 625.0
- (B) 400.0 725.0
- (C) 725.0 400.0
- (D) 400.0 400.0
- (E) 725.0 725.0

10. Consider the following code segment and class.

```
Bank tom = new Bank(500.0);
Bank sue = new Bank(tom);
sue.makeDeposit(225.0);
tom.makeWithdrawal(100.0);
System.out.println(tom.getBalance() + " " + sue.getBalance());
```

```
class Bank
{
    private double balance;

    public Bank(double amount)
    {
        balance = amount;
    }

    public Bank(Bank obj)
    {
        balance = obj.balance;
    }

    public double getBalance()
    {
        return balance;
    }

    public void makeDeposit(double amount)
    {
        balance += amount;
    }

    public void makeWithdrawal(double amount)
    {
        balance -= amount;
    }
}
```

What is printed as a result of executing the code segment?

- (A) 625.0 625.0
- (B) 400.0 725.0
- (C) 725.0 400.0
- (D) 400.0 400.0
- (E) 725.0 725.0

11. Consider the following code segment and class.

```
Game game = new Game();  
game.play();  
System.out.println(game.getScore());
```

```
class Game  
{  
    private int score;  
  
    public void play()  
    {  
        score = (int) (Math.random() * 10000);  
    }  
  
    public int getScore()  
    {  
        return score;  
    }  
}
```

What is printed by the code segment?

- (A) 0
- (B) a compile error message
- (C) an `IllegalCastException` error message
- (D) an integer value x , such that $0 \leq x < 10000$
- (E) an integer value x , such that $0 < x \leq 10000$

12. Consider the following code segment and class.

```
Game game1 = new Game(5);
game1.play();
System.out.println(game1.getScore());
Game game2 = new Game();
game2.play();
System.out.println(game2.getScore());
```

```
class Game
{
    private int score;
    private int level;

    public Game(int lev)
    {
        score = 0;
        level = lev;
    }

    public void play()
    {
        score = (int) (Math.random() * 10000);
        score = score / level;
    }

    public int getScore()
    {
        return score;
    }
}
```

What is printed as a result of executing the code segment?

- (A) 0
- (B) a compile error message
- (C) an `IllegalCastException` error message
- (D) an integer value x , such that $0 \leq x < 10000$
- (E) an integer value x , such that $0 < x \leq 10000$

Chapter 6 Questions

Static One-Dimensional Arrays

01. Consider the following method.

```
/** Precondition: n >= 2
 */
public static int method0601(int n)
{
    int[] temp = new int[n];
    temp[0] = 4;
    temp[1] = 7;
    for (int k = 2; k < n; k++)
        temp[k] = temp[k-1] + temp[k-2];
    return temp[n];
}
```

What value is returned by the call `method0601(8)`?

- (A) 47
- (B) 76
- (C) 123
- (D) 199
- (E) No value is returned due to an `ArrayIndexOutOfBoundsException` error

02. Consider the following method.

```
/** Precondition: list is a non-empty array.
 */
public static void method0602(int[] list)
{
    int max = list.length-1;
    for (int k = 0; k < max; k++)
        if (list[k] < list[k+1])
        {
            int temp = list[k];
            list[k] = list[k+1];
            list[k+1] = temp;
        }
}
```

Which of the following correctly describes the result of calling method `method0602`?

- (A) The smallest number is located in `list[max]`.
- (B) The smallest number is located in `list[max-1]`.
- (C) The largest number is located in `list[max]`.
- (D) The largest number is located in `list[max-1]`.
- (E) The elements in the `list` array are reversed.

03. Consider the following incomplete `getMean` method.

```
/**
 * Precondition: list is a non-empty array.
 * Postcondition: getMean returns the mean of the list array values.
 */
public static double getMean(int[] list)
{
    /* missing code */
}
```

Which of the following implementations of `/* missing code */` will make method `getMean` work as intended?

Implementation I

```
double temp = 0.0;
for (int k = 0; k < list.length; k++)
    temp += list[k];
return (temp / list.length);
```

Implementation II

```
int temp = 0;
for (int k = 0; k < list.length; k++)
    temp += list[k];
return (temp / list.length);
```

Implementation III

```
int temp = 0;
for (int k = 0; k < list.length; k++)
    temp += list[k];
return ((double) temp / list.length);
```

- (A) I only
- (B) II only
- (C) III only
- (D) I and II
- (E) I and III

04. Consider the following method.

```
/** Precondition: list is a non-empty array.
 */
public static void method0604(int[] list)
{
    for (int p = 0; p < list.length; p++)
    {
        int max = list.length - 1;
        int temp = list[p];
        list[p] = list[max-p];
        list[max-p] = temp;
    }
}
```

Which of the following correctly describes the result of calling method0604?

- (A) The elements in the list array are in random order.
- (B) The list array is sorted in descending order.
- (C) The list array is sorted in ascending order.
- (D) The elements in the list array are in reverse order.
- (E) The elements in the list array appear unchanged.

05. Consider the following method.

```
/** Precondition: list is a non-empty array.
 */
public static void method0605(int[] list)
{
    for (int p = 0; p < list.length/2; p++)
    {
        int max = list.length - 1;
        int temp = list[p];
        list[p] = list[max-p];
        list[max-p] = temp;
    }
}
```

Which of the following correctly describes the result of calling method0605?

- (A) The elements in the list array are in random order.
- (B) The list array is sorted in descending order.
- (C) The list array is sorted in ascending order.
- (D) The elements in the list array are in reverse order.
- (E) The elements in the list array appear unchanged.

06. Consider the following code segment.

```
int[] list = {5,10,15,20,25,20,15,10,5};
int max = list.length-1;
for (int k = max; k > 0; k--)
    list[k] = list[k] / list[max];
for (int k = 0; k < list.length; k++)
    System.out.print(list[k] + " ");
```

What will be printed as a result of executing the code segment?

- (A) 5 10 15 20 25 20 15 10 1
- (B) 1 2 3 4 5 4 3 2 1
- (C) 5 10 15 20 25 20 15 10 1
- (D) 1 1 1 1 1 1 1 1 1
- (E) 5 5 5 5 5 5 5 5 5

07. Consider the following code segment.

```
int[] list = {2,4,8,16,32,64,128,256};
for (int k = 1; k < list.length; k++)
    list[k] = list[k] / list[k-1];
for (int k = 0; k < list.length; k++)
    System.out.print(list[k] + " ");
```

What will be printed as a result of executing the code segment?

- (A) 1 1 1 1 1 1 1 1
- (B) 2 4 8 16 32 64 128 256
- (C) 1 2 4 8 16 32 64 128
- (D) 2 2 2 2 2 2 2 2
- (E) 2 2 4 4 8 8 16 16

08. Consider the following code segment.

```
int[] list1 = {2,4,8,16,32,64,128,256};
int[] list2 = list1;
int max = list1.length - 1;
for (int k = 0; k < list1.length ; k++)
    list2[k] = list1[max-k];
for (int k = 0; k < list1.length; k++)
    System.out.print(list1[k] + " ");
```

What will be printed as a result of executing the code segment?

- (A) 256 128 64 32 32 64 128 256
- (B) 256 128 64 32 16 8 4 2
- (C) 256 256 256 256 256 256 256 256
- (D) 2 2 2 2 2 2 2 2
- (E) 4 8 16 32 64 128 256 512

09. Consider the following code segment and method.

```
int[] list = {56,23,78,54,11,95,60,17,64};
list = mystery(list);
for (int item: list)
    System.out.print(item + " ");

/** Precondition: x is a non-empty array.
 */
public static int[] mystery(int[] x)
{
    int[] temp = new int[x.length];
    int q = temp.length - 1;
    for (int p = 0; p < x.length; p++)
    {
        temp[q] = x[p];
        q--;
    }
    return temp;
}
```

What will be printed as a result of executing the code segment?

- (A) 56 23 78 54 11 95 60 17 64
- (B) 11 17 23 54 56 60 64 78 95
- (C) 95 78 64 60 56 54 23 17 11
- (D) 64 17 60 95 11 54 78 23 56
- (E) An `ArrayIndexOutOfBoundsException` error message

10. Consider the following code segment.

```
int[] list = {56,23,78,54,11,95,60,17,64};
for (int item: list)
{
    item = 99;
    System.out.print(item + " ");
}
```

What will be printed as a result of executing the code segment?

- (A) 56 23 78 54 11 95 60 17 64
- (B) 64 17 60 95 11 54 78 23 56
- (C) 99 23 78 54 11 95 60 17 64
- (D) 56 23 78 54 11 95 60 17 99
- (E) 99 99 99 99 99 99 99 99 99

11. Consider the following code segment.

```
int[] list = {56,23,78,54,11,95,60,17,64};
for (int item: list)
    item = 99;
for (int item: list)
    System.out.print(item + " ");
```

What will be printed as a result of executing the code segment?

- (A) 56 23 78 54 11 95 60 17 64
- (B) 64 17 60 95 11 54 78 23 56
- (C) 99 23 78 54 11 95 60 17 64
- (D) 56 23 78 54 11 95 60 17 99
- (E) 99 99 99 99 99 99 99 99 99

12. Consider the following code segment and method.

```
int[] list1 = {56,23,78,54,11,95,60,17,64};
int[] list2 = {32,44,87,11,90,56};
swap(list1,list2);
for (int item1: list1)
    System.out.print(item1 + " ");
System.out.println();
for (int item2: list2)
    System.out.print(item2 + " ");

public static void swap(int[] l1, int[] l2)
{
    int[] l3 = l1;
    l1 = l2;
    l2 = l3;
}
```

What will be printed as a result of executing the code segment?

- (A) 56 23 78 54 11 95 60 17 64
32 44 87 11 90 56
- (B) 32 44 87 11 90 56
56 23 78 54 11 95 60 17 64
- (C) 56 23 78 54 11 95 60 17 64
56 23 78 54 11 95 60 17 64
- (D) 32 44 87 11 90 56
32 44 87 11 90 56
- (E) An `ArrayIndexOutOfBoundsException` error message