

Test 1 study guide

All chapters 1- 6 are included on the test.

- Decision making (if, switch, conditional statement)
- Loops (while, for, do-while, for-each)
- Arrays
- ArrayList
- Methods / functions
- Classes (system : String, Scanner,... and User defined Type: Student, Car,...)
- Driver classes (includes main method)

To practice do the problems at the back of each section and chapter, review your assignments. Here are some sample questions as well:

1. Expressions

For each expression in the left-hand column, indicate its value in the right-hand column. Be sure to list a constant of appropriate type (e.g., 7.0 rather than 7 for a double, Strings in quotes, true/false for a boolean).

<u>Expression</u>	<u>Value</u>
$3 * 4 + 5 * 6 + 7 * -2$	<u>$12 + 30 + (-14) = 28$</u>
$1.5 * 2.0 + (5.5 / 2) + 5 / 4$	<u>$3.0 + 2.75 + 1 = 6.75$</u>
$"1" + 2 + 3 + "4" + 5 * 6 + "7" + (8 + 9)$	<u>123430717</u>
$1 / 2 > 0 \parallel 4 == 9 \% 5 \parallel 1 + 1 < 1 - 1$	<u>$0 > 0 \parallel 4 == 4 \parallel 2 < 0 \rightarrow \text{false} \parallel \text{true} \parallel \text{false} \rightarrow \text{true}$</u>

2. Parameter Mystery

```
public class ParameterMystery {
    public static void main(String[] args) {
        String x = "java" , y = "tyler" , z = "tv";
        String rugby = "hamburger";
        String ava = "donnie";

        hamburger(x, y, z);
        hamburger(z, x, y);
        hamburger("rugby", z, hello);
        hamburger(y, rugby, "x");
        hamburger(y, y, "ava");
    }
    public static void hamburger(String y, String z, String x) {
        System.out.println(z + " and " + x + " like " + y);
    }
}
```

Output:

```
tyler and tv like java
java and tyler like tv
***** ERROR hello is not declared
hamburger and x like tyler
tyler and ava like tyler
```

3. If/Else

For each call of the method below, write the value that is returned:

```
public static int mystery(int a, int b) { int c;
```

```
    if (a > b) {
```

```
        c = a;
```

```
    } else if (b % a == 0)
```

```
    { c = b;
```

```
    } else {
```

```
        c = b + (a - (b % a));
```

```
    }
```

```
    return c;
```

```
}
```

Method Call

Value Returned

mystery(4, 2)

4

mystery(5, 4)

5

mystery(5, 13)

15

mystery(5, 17)

20

mystery(4, 8)

8

4. While Loop

For each call of the method below, write the output that is printed:

```
public static void mystery(int i, int j) { while (i != 0
```

```
    && j != 0) {
```

```
        i = i / j;
```

```
        j = (j - 1) / 2;
```

```
        System.out.print(i + " " + j + " ");
```

```
    }
```

```
    System.out.println(i);
```

```
}
```

Method Call

Output

mystery(5, 0);

5

mystery(3, 2);

1 0 1

mystery(16, 5);

3 2 1 0 1

mystery(80, 9);

8 4 2 1 2 0 2

mystery(1600, 40);

40 19 2 9 0 4 0

5. Trace the following code segment and show what is printed on the monitor:

```
public static int mystery(int x) {  
    int y = 1 , z = 0;  
    while (y <= x) {  
        y = y * 10;  
        z++;  
    }  
    z--;  
    System.out.println(y + z + x);  
    return z;  
}
```

```
int    a = 40 , b;
```

```
b = mystery(a);  
System.out.print(a + b + (a+b) );
```

output:

```
141  
82
```

6. Write a static method named hasMidpoint that accepts three integers as parameters and returns true if one of the integers is the midpoint between the other two integers; that is, if one integer is exactly halfway between them. Your method should return false if no such midpoint relationship exists. The integers could be passed in any order; the midpoint could be the 1st, 2nd, or 3rd. You must check all cases.

Calls such as the following should return true:

```
hasMidpoint(4, 6, 8)  
hasMidpoint(2, 10, 6)  
hasMidpoint(8, 8, 8)  
hasMidpoint(25, 10, -5)
```

Calls such as the following should return false:

```
hasMidpoint(3, 1, 3)  
hasMidpoint(1, 3, 1)  
hasMidpoint(21, 9, 58)  
hasMidpoint(2, 8, 16)
```

```
public static boolean mystery(int x, int y, int z) {  
    double    mid = (x+y+z)/3 ;  
    if (mid == x || y == mid || z == mid)  
        return true;  
    else  
        return false ;  
}
```

4. Write a static method named `sequenceSum` that prints terms of the following mathematical sequence:

$$1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \dots + \frac{1}{N} \quad \text{also written as: } \sum_{i=1}^N \frac{1}{i}$$

Your method should accept a real number as a parameter representing a limit, and should add and print terms of the sequence until the sum of terms meets or exceeds that limit. For example, if your method is passed 2.0, print terms until the sum of those terms is at ≥ 2.0 . You should round your answer to 3 digits past the decimal point.

The following is the output from the call `sequenceSum(4)`; $1 + 1/2 + 1/3 + 1/4 = 2.083$

Calls	<code>sequenceSum(0);</code>	<code>sequenceSum(1);</code>	<code>sequenceSum(2);</code>
Output	0.0	1 = 1.000	1 + 1/2 = 1.500
Call	<code>sequenceSum(8);</code>		
Output	$1 + 1/2 + 1/3 + 1/4 + 1/5 + 1/6 + 1/7 + 1/8 = 2.718$		

```

public static void sequenceSum(double n){

    double sum = 0;
    for (int I = 1 ; I <= n ; I++){
        System.out.print(I + "/" + n + " + ");
        sum += 1.0/I ;
    }
    if (n == 0)
        System.out.println(n);
    else
        System.out.println(" = " + sum);
}

```

7. What's wrong with the following program?

```

public class SomethingIsWrong {
    public static void main(String[] args) {
        Rectangle myRect; // = new myRect(); need to instantiate the object
        myRect.width = 40; //use the setter if private data member
        myRect.height = 50; //use the setter if private data member
        System.out.println("myRect's area is " + myRect.area());
    }
}

```

8. What is the output of the following code segment?

```

String[] students = new String[10];
String studentName = "Peter Smith";
students[0] = studentName;
studentName = "joe";
System.out.println(students[0] + "\t" + studentName); → Peter Smith    Joe

```

Questions 9 – 11

9. Develop a UML for a class to represent a Rectangle, and then develop the class in Java.

RectangleType
- width : double - length: double
+ RectangleType() + RectangleType(double , double) + setWidth(double) : void + setLength(double) : void + SetDimension(double l, double w): void + getLength() : double + getWidth() : double - area() : double - perimeter() : double + print() : void + toString() : String +equals(RectangleType): boolean

```
public class RectangleType {
    private double    width, length ;

    public RectangleType(double l, double w){
        SetDimension(l, w);
    }

    public RectangleType(){
        length = 0;
        width = 0;
    }

    public void setWidth(double wd){
        if (wd >= 0)
            width = wd;
        else
            width = 0;
    }

    public void setLength(double length){
        if (length >= 0)
            this.length = length;
        else
            this.length = 0;
    }
}
```

```

public void SetDimension(double l, double w){
    SetWidth(w);
    SetLength(l);
}
public double getLength(){
    return length;
}

Public double getWidth(){
    return width;
}

private double Area() {
    return length * width;
}

private double Perimeter() {
    return 2 * (length + width);
}

public void print() {
    System.out.println("Length = " + length + "\nWidth = " + width);
}

public String toString() {
    String Temp = "Length = " + length + "\nWidth = " + width;
    Temp += "\nArea = " + Area();
    Temp += "\nPerimeter = " + Perimeter() + "\n";
    return Temp;
}

public boolean equals(RectangleType obj){
    if (length == obj.getLength() && width == obj.getWidth())
        return true;
    return false ;
}
}

```

10. Write a driver to use Rectangle class developed for previous question and test all the methods.

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

```
    RectangleType R1, R2;
```

```
    R1 = new RectangleType();
```

```
    R2 = new RectangleType(4, 5);
```

```
    R1.print();
```

```

    R2.print();
    System.out.println("\nRectangle 1\n"+R1);
    System.out.println("\n"+ R1.toString());
    String s1 = "Rectangle 2 is " + R2.toString();
    System.out.println(s1);
}

```

11. Write a java code segment to declare an array of size 10 of type Rectangle and read data for them from keyboard.

```

RectangleType [] rList = new RectangleType[10] ;
double l, w ;
for (int I = 0 ; I < 10 ; I ++){
    System.out.print("Enter width : ")
    w = KB.nextDouble();
    System.out.print("Enter length : ")
    l = KB.nextDouble();

    rList [ I ] = new Rectangle(l,w);
}

```

12. Write a java code segment to declare an ArrayList of type Rectangle and read data for 3 rectangle from keyboard to store in the list.

// Create new ArrayList.

```

ArrayList<Rectangle> elements = new ArrayList<>();

```

// Add three elements.

```

for (int I = 0 ; I < 3 ; I ++){
    Rectangle R1 = new Rectangle();
    R1.read();
    elements.add(R1);
}

```