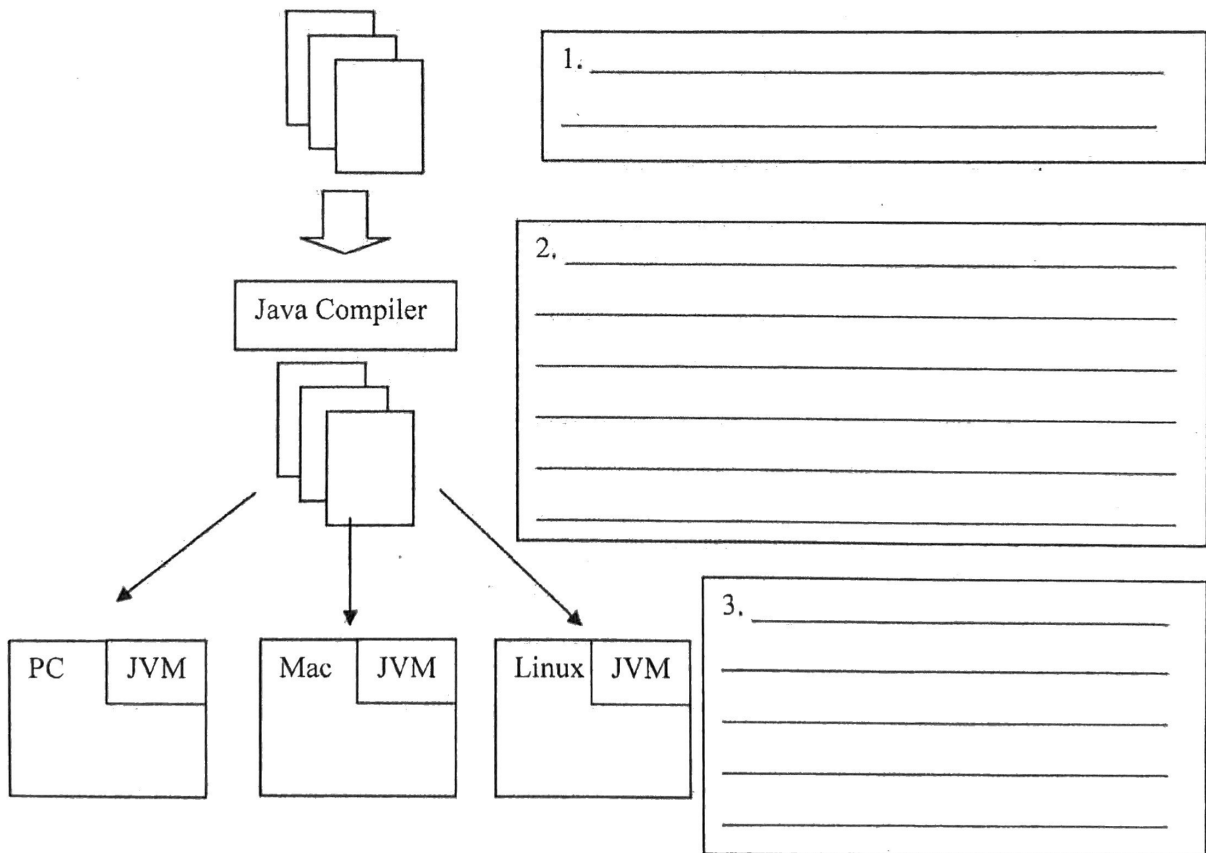


Compiling and Running Java Programs.

A compiler is a program that translates code to machine language that the central processing unit (the computer's "brain") can understand. In many languages, the source code must be customized for different operating systems and then compiled for certain types of processors.

Java works as follows:



The JRE (Java Runtime Environment) includes the JVM and java standard class files. It is the JRE that you see occasionally updated on your computer.

AP Computer Science A

Unit 3. Boolean Expressions. Notes

Comments are statements that help describe a program but are ignored by the computer. There are two types of comments.

1) **Single line comments.** Anything to the right of `//` is a comment.

```
double rate;           // text that describes the variable's purpose
```

2) **Multiple-line comments.** Anything between a `/*` and the closing `*/` is a comment.

```
/* This program was written by Jennifer Somebody  
   August 2019.      */
```

Types of Errors.

Compiler Errors aka _____ **Errors.** A compiler is software that checks if a program correctly follows a programming language's rules. For example, are there semicolons where they are supposed to be? Are only integer values being assigned to ints and so on. Compilers do other things, but this is all we are concerned about here.

Example 1.

```
int x = 8.9;           // compiler error
```

In Java, a program must be free of compiler errors or _____

Run-time Errors occur while your program is running. They may be due to something the user does or some flaw in the code that is only exposed when it runs.

Example 2. Given the following code is part of a larger program and compiles, a run-time error will occur if _____

```
Scanner s = new Scanner( System.in );  
System.out.println( "Enter a number" );  
int n = s.nextInt();
```

Example 3. The following code compiles because the expression on the right evaluates to an int. However, a run-time error occurs because _____

```
int x = 10 / 0;  
System.out.println( x );
```

Logic Errors occur when your code compiles and runs but the output is incorrect. These are generally the _____

Boolean Expressions. A Boolean expression is either _____ or _____.

For example: `x > 8`

If `x` is greater than or equal to 8, then the expression is true. Otherwise the expression is false.

Here are 6 operators used in Boolean expressions _____

1. If <code>num</code> is 10, what is the value of this expression?	<code>num >= 10</code>
2. Give a value for <code>y</code> that makes this expression false?	<code>y != 5</code>
3. Give a value for <code>y</code> that makes this expression true?	

if and if-else Statements. These statements are called control statements because they control whether a block of code is executed or not. Some texts refer to them as conditionals because they represent a condition that may be true or false.

Example 1.

```
int x = (int) ( 10 * Math.random() ); _____
if ( x > 4 ) {
    x++;
    System.out.print( x );
}
System.out.println( "Done" );
```

The if must be followed by parentheses.
The () must contain _____

If the block is more than one statement, _____

In the above example, if `x` has a value of 7, what is printed? _____

Example 2.

```
int z = (int) ( 3 * Math.random() ); _____
if ( z == 2 )
    z++;
else _____
    z--;
System.out.println( z );
```

In this example, the curly braces were NOT required because the block of code following the if statement and the code following the *else* keyword _____

List the numbers that might be printed in the above example. _____

Do exercises 1 to 19.

Block Scope. If a variable is declared with a block of code, _____

Example of code with a compiler error due to block scope.

```
int a = (int)(10*Math.random()) + 1;
if ( a >= 5 ){
    int b = a + 10;
    System.out.println( a + b );
}
System.out.println( a + b ); // compiler error: cannot find symbol
```

If you need to use a variable within a block AND use it after the block, then declare it before the block.

One possible alternative approach that does NOT generate a compiler error.

```
int a = (int)(10*Math.random()) + 1;
int b = 0;
if ( a >= 5 ){
    b = a + 10;
    System.out.println( a + b );
}
System.out.println( a + b ); // compiler error: cannot find symbol
```

if-else if Statements. The basic if - else structure can be expanded to handle multiple conditionals using else-if statements.

Important. You only execute the code associated with the _____

Example.

```
int h = (int)( 50 * Math.random() );
if ( h % 7 == 0 )
    System.out.print( "R" );
else if ( h < 22 )
    System.out.print( "O" );
else if ( h > 10 )
    System.out.print( "C" );
else
    System.out.print( "K" );
```

If h is 14, what is printed? _____

If h is 19, what is printed? _____

What value(s) of h will cause R to be printed? _____

Note. Not every *if*, or *else if*, needs to end with an *else*.

Do exercises 20 to 33.

Logical Operators. More complicated Boolean expressions can be composed using logical operators `&&` (AND) and `||` (OR)

`x > 10 && x < 15` is true when _____

`y < 0 || y > 100` is true when _____

1. What values of <i>b</i> will cause S to be printed?	// b is an int if (b >= 3 && b < 7) System.out.println("S");
2. What values of <i>c</i> will cause V to be printed?	// c is an int if (c >= 3 c < 7) System.out.println("V");
3. What values of <i>d</i> will cause Z to be printed?	// d is an int if (d > 0 && d > 7) System.out.println("Z");

There is a third logical operator that is useful in certain situations: the NOT operator _____

boolean a = !true; _____

boolean z = !false; _____

Nested If Statements. When there is an if statement within another if statement, that is known as *nested if* statements.

Example.

```
int k = (int) (15 * Math.random());
if (k % 2 == 0) {
    if (k % 3 == 0)
        System.out.println("A");
    else
        System.out.println("B");
} else
    System.out.println("C");
```

For what values of *k* is A printed? _____

For what values of *k* is B printed? _____

For what values of *k* is C printed? _____

Comparing String Objects. A compiler error occurs if you try to compare objects using _____ . And usually, you will not get the result you want if you use _____ to compare objects. (We will discuss why this is in unit 5.) To compare strings, we will use the _____ and _____ methods.

Do exercises 34 to the end.