Standard 1 – Programming environment

1. Name and explain 1 hardware term and 1 software term you are least familiar with.
   Hardware: optical drive, a disc drive that uses laser light or electromagnetic waves to read or write data to or from discs. Examples are CDs, DVDs, and Blu-ray disks.

   Software: bit and byte, a bit is the smallest unit of information and a byte is 8 bits.
2. What does a break point do?
   A break point runs a program up to that point then stops during debugging.
3. When debugging, what are you normally monitoring?
   You are normally monitoring the variables during debugging.
4. What does it mean to "step through" an application?
   Go to the next breakpoint when debugging an application.
5. Describe and give an example of a syntax error.
   An error that violates the conventions of the programming language, and causes the program to not execute. Spelling variable and function names wrong, forgetting a semicolon, mismatched parentheses/brackets.

   ```
   System.out.println("Hello"))
   ```
6. Describe and give an example of a run-time error.
   An error that occurs during the execution of the program. Examples: Dividing by 0 error, IO errors, exceptions.
7. Describe and give an example of a logic error.
   The program will still compile and execute, but the output of the program is wrong. Examples: using the assignment operator when you meant to use the comparison operator, multiplying two numbers instead of adding, etc.
8. When an application is compiled, what is it changed to?
   When an application is compiled, it is changed to bytecode or an executable file.
9. Name and describe the terms/concepts you are least familiar with in Standard 1.
   Interpreted vs. compiled language: Interpreted languages are translated to something that an interpreter has to translate for a particular computer, while compiled languages are directly translated into machine code and no program is needed to translate.

Standard 2 – Programming methodology

1. Name and give examples of 3 string literals in your language.
   'Hello world!'
2. Describe the difference between an operator and operand.
   Operators change variables by adding, subtracting, etc., operands are the variables that are being changed. There are binary and unary operators.
3. Write the line of code that constitutes your language "entry point."

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

```
        }
}
```

4. Name and describe what it means to plan an app including placeholder for named classes and functions/methods.
   Placing the methods and classes according to your plan even if it doesn't have logic yet.
5. Write a method in your language that takes in a decimal value and converts it from Fahrenheit to Celsius and returns the new value.
   ```
   static double convertToCelsius(double f) {
           double celsius = (f – 32.0) * (5.0 / 9.0);
           return celsius;
   }
   ```
6. Site a good and bad example of a number variable, a method name and a class name.
   Number variable – good: radius, bad: num.
   Method name – good: calculateVolume(), bad: volume()
   Class name – good: Sphere, bad: sphere
7. Site how single line comments and multiple line comments are accomplished in your language.
   Use // for single line comments and /**/ for multiple line comments.
8. Name and describe the terms you are least familiar with in Standard 2.
   String literal:

Standard 3 – Language fundamental commands and operations

1. Write a sample line of code naming and initializing each of your language's primitive data types.

```
byte a = 0;
short b = 0;
int c = 0;
long d = 0;
float e = 0;
double f = 0;
char g = 'a';
boolean h = true;
```

2. Write a line of code takes information from the user and assigns it to a named String variable.

```
String input = scanner.nextLine();
```

3. Write a line of code that casts a number as a String.

```
int number = 92;
stringNumber = Integer.toString(number);
```

4. List and describe all known operators in your language.

```
// Arithmetic operators (+, -, *, /, ++, %): Performs math on variables.
// Assignment operators (+=, -=): Used to assign values to variables.
// Comparison operators (==, !=, <, <=): Used to compare values.
// Boolean operators (||, &&, !): Used in boolean expressions.
```

5. Write a line of code setting a reference variable.

```
Scanner scanner = new Scanner(System.in);
```

6. Name and describe the terms/concepts you are least familiar with in standard 3.
   Enumerators: A special “class” that represents a list of constants. Use enum keyword and a name for it, then inside the brackets separate the constants with a comma.

Standard 4 – Control structures

1. Site all of your language’s logic operators.
   ||, &&, !
2. Site all of your language’s relational operators.
   <, >, <=, >=, ==, !=
3. Write a complex if statement checking “or” conditions.

```
if (x < 0 || !run || input.equals("i")) {
    System.out.println("game over");
    }
```

4. Write an if-else statement checking “and” conditions.

```
if (x < width && x > 0 || y < height && y > 0) {
    System.out.println("true");
} else {
    System.out.println("false");
}
```

5. Write a sample switch, case, default statement.

```
switch (numOfSides) {
    default:
         name = "null";
         break;
   case 3:
         name = "triangle";
         break;
   case 4:
         name = "square";
         break;
   case 5:
         name = "pentagon";
         break;
   case 6:
         name = "hexagon";
         break;
   }
```

6. Write a for loop that will print numbers 1-100 to console.

```
for (int i = 1; i <= 100; i++) {
      System.out.println(i);
}
```

7. Write a sample while or do-while statement.

```
while (i > 0) {
      run = true;
      i--;
}
```

8. What is a nested loop, give example.
   A loop inside a loop, such as an if statement inside an if statement.
9. Describe a strategy for keeping a running total in an app.
   Declare a global variable, each time something happens, such as an iteration of a loop, add a number to the variable.
10. Name and describe the terms/concepts you are least familiar with in Standard 4.
    Scope of an identifier: The part of the program where a variable is accessible.

824 Computer Programming 1B (Java)

Test 827: 51 Questions for 60 Points | Test Available April 21 to May 30

	St.1	St.2	St.3	St.4	St.5	St.6	St.7	St.8	St.9	St.10
# of Questions	7	2	14	11	N/A	1	10	7	6	2

STANDARD 1: Students will be familiar with and use a programming environment.

Objective 1: Demonstrate knowledge of external and internal computer hardware.

1. Describe the functions of basic external computer hardware devices (monitor, printer, keyboard, mouse, adapters, other devices).
2. Describe the functions of the internal components of computers (CPU, RAM, ROM, motherboard, graphics card, hard drive, optical drive).
3. Understand what a bit and a byte is and how it relates to memory storage.

Objective 2: Demonstrate knowledge of software concepts.

1. Define the distinction between computer software and hardware.
2. Identify software categories such as application software, web-based software, or OS.
3. Describe the difference between an interpreted language vs a compiled language.

Objective 3: Demonstrate the ability to compile, debug, and execute programs.

1. Demonstrate how to use an editor/IDE to compile and run programs.
2. Understand the difference between syntax, run-time, and logic errors.
3. Demonstrate how to debug programs.
4. Optional -- Use a debugger to set break-points, and step through code to track down errors at runtime.

STANDARD 2: Students will employ accepted programming methodology.

Objective 1: Demonstrate the ability to use good programming style.

1. Demonstrate how to use white space properly.
2. Employ an appropriate naming convention.
3. Construct identifiers with meaningful format (ie: camelCase, under_scores,PascalCase, and ALLCAPS).

Objective 2: Understand that software development is a process and use a variety of creation techniques to develop 21st Century Skills.(www.p21.org)

1. Understand specifications and requirements for computer programs.
2. Decompose the problem into appropriate components.
3. Design solutions using algorithms and other problem solving techniques..
4. Write the code for a program.
5. Test programs for errors and proper functionality.
6. Provide internal and external documentation for a program during development.
7. Redo all steps as needed.

Objective 3: Identify the syntactical components of a program.

1. Identify keywords, identifiers, operators, operands, and literals.
2. Identify the entry-point of a program.
3. Identify statements and expressions in a program.
4. Identify program components such as functions, methods, or procedures.

1.1 Name and explain 1 hardware term and 1 software term you are LEAST familiar with.
1.2 What does a break point do?
1.3 When debugging, what are you normally monitoring?
1.4 What does it mean to "step through" an application?
1.5 Describe and give an example of a syntax error.
1.6 Describe and give an example of a run--time error.
1.7 Describe and give an example of a logic error.
1.8 When an application is compiled, what is it changed to?
1.9 Name and describe the terms/concepts you are least familiar with in Standard 1 (at least one).

2.1 Name and give examples of 3 string literals in your language.
2.2 Describe the difference between an operator and an operand.
2.3 Write the line of code that constitutes your language "entry point."
2.4 Name and describe what it means to plan an app including placeholders for named classes and functions/methods.
2.5 Write a method in your language that takes in a decimal and converts it from Fahrenheit to Celsius and returns the new value.
2.6 Site a good and bad example of a number variable, a method name and a class name.
2.7 Site how single line comments and multiple line comments are accomplished in your language.
2.8 Name and describe the terms/concepts you are least familiar with in Standard 2 (at least one).

STANDARD 3: Students will properly use language-fundamental commands and operations.

Objective 1: Demonstrate the ability to use basic elements of a specific language.

1. Write programs formatted based on the conventions of the utilized language.
2. Declare, initialize, and assign values to constants and variables.
3. Demonstrate the ability to use input and output commands.
4. Communicate clearly with output values stored in identifiers.(www.p12.org)
5. Demonstrate the ability to use strings in programs.

Objective 2: Employ basic arithmetic expressions in programs.

1. Use basic arithmetic operators (modulus, multiplication, division, addition, subtraction).
2. Understand order of operation of expressions.
3. Write expressions that mix floating-point and integer expressions.

Objective 3: Demonstrate the ability to use data types in programs.

1. Declare and use variable types (primitives, reference, or object).
2. Declare and use constants.
3. Know the difference between data types and their application (boolean, integer, floating point, strings).
4. Optional -- Declare and use enumerators as a list of constants.

STANDARD 4: Students will properly employ control structures.

Objective 1: Demonstrate the ability to use relational and logical operators in programs.

1. Compare values using relational operators.
2. Form complex expressions using logical operators.

Objective 2: Demonstrate the ability to use decisions in programs.

1. Employ simple IF structures.
2. Use IF-ELSE structures.
3. Write programs with nested IF-ELSE structures.
4. Make multiple-way selections (switch, case).

Objective 3: Demonstrate the ability to use loops in programs.

1. Use initial, terminal, and incremental values in loops.
2. Construct while, do-while, and for loops
3. Describe the various ways that loops can end.
4. Utilize nested loops.
5. Explain how to avoid infinite loops.
6. Accumulate running totals using loops.

Objective 4: Demonstrate the ability to use modularity in programs using functions or methods.

1. Demonstrate how to use language-defined components.
2. Utilize value and reference parameters.
3. Understand the scope of identifiers (local, class variables).
4. Return values.

3.1 Write a sample line of code naming and initializing each of your languages' primitive data types.

3.2 Write an line of code takes information from the user and assigns it to a named String variable.

3.3 Write a line of code that casts a number as a String.

3.4 List and describe all known operators in your language.

3.5 Write a line of code setting a reference variable.

3.6 Name and describe the terms/concepts you are least familiar with in Standard 3 (at least one).

Boolean Logic:

4.1 Site all of your languages logic operators.

4.2 Site all of your languages relational operators.

4.3 Write a complex if statement checking "or" conditions.

4.4 Write an if else statement checking "and" conditions

4.5 Write a sample switch, case, default statement.

Iteration:

4.6 Write a for loop that will print numbers 1-100 to console.

4.7 Write a sample while or do while statement.

4.8 What is a nested loop; give example.

4.9 Describe a strategy for keeping a running total in an app.

4.10 Name and describe the terms/concepts you are least familiar with in Standard 2 (at least one).

STANDARD 5: Students will demonstrate knowledge of current ethical issues dealing withcomputers and information in a global society using 21st Century Skills.

Objective 1: Understand ethical responsibility of software developers

1. Explain the ethical reasons for creating reliable and robust software.
2. Explain the impact software can have on society.
3. Show how security concerns can be addressed in a program.

Objective 2: Demonstrate knowledge of the social and ethical consequences of computers.

1. Describe how computer-controlled automation affects a workplace and society.
2. Explain the ramifications of society's dependence on computers.
3. Use 21st Century Skills to understand and address global issues

1. Explain how computers can compromise privacy.
2. Exhibit knowledge of privacy laws.
3. Describe responsibilities of people who control computer information.

Objective 4: Demonstrate knowledge of computer, information and software security.

1. Exhibit knowledge of copyright laws.
2. Explain how computers could erroneously be used to compromise copyright laws.
3. Give examples of ways to protect information on computer systems.
4. Identify ways to protect against computer virus