**Standard 1**

**1.1 Name and explain 1 hardware term and 1 software term you are LEAST familiar with.**

- One hardware term I’m unfamiliar with is the main memory. The main memory is the largest RAM in a memory hierarchy in a computer system

- One software term I’m unfamiliar with is API which stands for application programming interface. The API is a set of commands, functions, protocols, and objects that programmers can use to create software or interact with an external system.

**1.2 What does a breakpoint do?**

- Breakpoints are basically markers for where the program when run or executed should stop. This is used when debugging a program or application.

**1.3 When debugging, what are you normally monitoring?**

- You are monitoring the application and looking for a spot or line of code that could potentially have errors that could ruin your code or make the code not runnable.

**1.4 What does it mean to “step through” an application?**

- It means going through the code line by line when debugging and have a breakpoint in the program.

**1.5 Describe and give an example of a syntax error.**

- Syntax errors can be done like when using a different language code like python when coding a java application. Another more specific example can include making a boolean but not giving it true or false returning values

**1.6 Describe and give an example of a run-time error.**

- Run-time error is an error that happens while the program is running. An example of a run time error can be done with bad or undefined math errors:

Ex:

Int a = 1

Int b = 10

Int c = 10

Int x = a / (b - c); -> this is a run time error because while the code itself doesn’t have any problems, when the code is run the x = 1/0 which is not a number. It’s undefined

**1.7 Describe and give an example of a logic error.**

- a logic error is a bug in the program that causes it to operate incorrectly but not to terminate abnormally or to crash. Ex: int LetterCount = 0;

**1.8 When an application is compiled, what is it changed to?**

- For a java machine to be able to run a Java program, the program’s Java source code must be compiled into byte-code using the javac compiler. Java byte-code is a platform independent version of machine code.

**1.9 Name an describe the terms/concepts you are least familiar with in Standard 1 (at least one).**

- I’m not very familiar with how to tell the difference between say a logic error and a runtime error.

**Standard 2**

**2.1 Name and give examples of 3 string literals in your language.**

- one example include a java integer literal. There are octal literals, hexadecimal literals, and decimal integer literals. An example of an octal literal: int octLit = 0400; which is the octal equivalent of decimal 256

- Then there are also java floating-point literals. Ex: float f = 89.0; float ff = 89.0f;

- Another one is a java boolean literal which only have two possible values: true or false. Ex: boolean boolFalse = false;

**2.2 Describe the difference between an operator and an operand.**

- Operands are numeric, text and Boolean values that a program can manipulate and use as logic guidelines for making decisions. Operands include mathematical actions like plus, minus, multiply and divide for manipulating data and changing numerical values.

- Operators are used to manipulate and check operand values.

**2.3 Write the line of code that constitutes your language “entry point.”**

- public static main void(String[])

- private static main void(String[])

**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 a new value.**

import java.util.Arrays;

import java.util.Scanner;

public class FtoC {

public static void main(String args[]) {

Scanner cmd = new Scanner(System.in);

System.out.println(“Enter Temp in F: “);

float temperature = cmd.nextFloat();

float celsius = toCelsius(temperature);

System.out.printf(“%.02f degree F temperature is equal to %.02f degree C);

System.out.println(Enter temp in degree C: “);

temperature = cmd.nextFloat();

float fahrenheit = toFaherenheit(temperature);

System.out.printf(“%.02f degree D is equal to %.02f degree F);

}

public static float toFahrenheit(float celsius) {

float fahrenheit = 9 \* (celsius / 5) + 32;

return fahrenheit;

}

public static flaot toCelsius(float fahrenheit) {

float celsius = (fahrenheit - 32) \* 5 / 9;

return celsius;

}

}

**2.6 Site a good and bad example of a number variable, a method class and a class name.**

-

**2.7 Site how single line comments and multiple line comments are accomplished in your language.**

- // (text) - for single line comments

- /\*text

text

text\*/ - for multiple line comments

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

**-** Entry point: the main method is the entry point of a Java application

**-** identifiers

**Standard 3**

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

- byte: byte b = 100;

- short: short s = 10000;

- int: int i = 100000;

- long: long SocialSecurityNumber: 999\_99\_9999L;

- float: float f1 = 123.4f;

- double: double d1 = 123.4; double d2 = 1.234e2;

- boolean: boolean result = true;

- char: char capitalC = ‘C’;

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

- String myString2 = newString();

myString = myString2;

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

- String str = “”;

int i = 5;

str = str + i;

**3.4 List and describe all known operators in your language.**

= Simple assignment operator

+ Additive operator (also used for String concatenation)

- Subtraction operator

\* Multiplication operator

/ Division operator

% Remainder operator

+ unary plus operator; indicates positive value (#s are positive w/out this, however)

- unary minus operator; negates an expression

++ increment operator; increments a value by 1

-- Decrement operator; decrements a value by 1

! Logical complement operator; inverts the value of a boolean

== equal to

!= not equal to

> greater than

>= greater than or equal to

< less than

<= less than or equal to

&& conditional-AND

| | Conditional-OR

?: ternary (shorthand for if-then-else statement)

instanceof - compares an object to a specified type

~ unary bitwise complement

<< signed left shift

>> signed right shift

>>> unsigned right shift

& bitwise AND

^ bitwise exclusive OR

| bitwise inclusive OR

**3.5 Write a line of code setting a reference variable.**

- String s1 = “ab”;

String s2 = s1;

s1 = s1 + “c”;

System.out.println(s1 + “ “ + s2);

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

- not really sure what floating-point and integer expressions are

- different variable types and how to distinguish the difference between them

**Standard 4**

**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**

if(str != null && str.isEmpty()) {

doSometingWith(str.charAt(0));

}

**4.4 Write a complex if else statement checking “and” conditions.**

if(str = null | | str.isEmpty()) {

complainAboutUnusableString();

}

**4.5 Write a sample switch, case, default statement.**

public class Switch {

public static void main(String[] args) {

int month = 8;

String monthString;

switch (month) {

case 1: monthString = “January”;

break;

case 2: monthString = “February”;

break;

case 3: monthString = “March”;

break;

case 4: monthString = “April”;

break;

case 5: monthString = “May”;

break;

case 6: monthString = “June”;

break;

case 7: monthString = “July”;

break;

case 8: monthString = “August”;

break;

case 9: monthString = “September”;

break;

case 10: monthString = “October”;

break;

case 11: monthString = “November”;

break;

case 12: monthString = “December”;

break;

default: monthString = “Invalid month”;

break;

}

System.out.prinln(monthString);

}

}

**Iteration:**

**4.6 Write a loop that will bring numbers 1-100 to console.**

class Loop {

public static void main(String[] args) {

for(int i=0; i<= 100; i++) {

System.out.println(i);

}

}

}

**4.7 Write a sample while or do while statement.**

While statement:

class While {

public static void main(String[] args) {

int count = 1;

while (count < 11) {

System.out.println(“count = “ + count);

count++;

}

}

}

Do-While Statement:

class DoWhile {

public static void main(String[] args) {

int count = 1;

do {

System.out.println(“count = “ + count);

} while (count < 11);

}

}

**4.8 What is a nested loop; give example.**

- A nested loop is basically one loop that is used inside another loop like a for loop in another for loop

for (int i = 1; i <= 5; i++) {

for (j = 1; j <= 10; j++) {

System.out.println((i \* j) + “ “);

}

System.out.println();

}

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

- to keep a running total in an app, a good way to do this would be to use the “Store” syntax. So you’d have to make a Store class with instances.

**4.10 Name and describe the terms/concepts you are least familiar with in Standard 4 (at least one).**

- i don’t know how keep a running total in an app while using loops but i can with a class

- i also don’t know how to do objective 4 part 1 and 2 on the standards list on the powerpoint