Introduction to Computer Science Waivers

I. All students must complete the “East Brunswick Public Schools Online Waiver Form.” Students can learn more about this form from their guidance counselor.

II. In order to waive Introduction to Computer Science (Intro) students must learn the concepts covered in Intro (the topics are listed below); they can accomplish this by:

A. Taking an online course

B. Studying on their own (books, online tutorials, etc.)

C. Taking a summer course (this course must be preapproved – there is a separate form that needs to be filled out.)

D. Any other means to learn the necessary topics/skills

III. All students need to put together a portfolio that demonstrates that they are proficient in the Intro topics.

A. Write a program that fulfills many of the concepts learned in Introduction to Computer Science (below).

B. Present the completed program to the APCS instructor in ***mid-June***.

* 1. This presentation will be part interview to determine if the student is prepared for APCS A, and will last about 20 minutes.

IV. The following is a brief list of resources that a student can use to prepare for the placement assessment.

[*http://www.learnjavaonline.org/*](http://www.learnjavaonline.org/)

[*http://www.codecademy.com/*](http://www.codecademy.com/) (This is a JavaScript tutorial, but is close enough)

<https://www.edx.org/course/introduction-programming-java-part-1-uc3mx-it-1-1x#.VLkXUr5gCoU> (This is a MOOC sponsored by Harvard and MIT)

Middlesex County College courses

CTE course

JAVA programming books that can be found that cover the topics

***V. Completing the waiver process does NOT guarantee a seat in the course.***

**Topics Covered in Intro to Comp. Sci.**

**Unit 1: Introduction to Computers and Programming**

* Describe the hardware and software components of a typical computer system
* Understand the history of computing and computers
* Understand the terminology of Computer Science
* Use an IDE with an editor and a compiler to create programs to execute on a computer
* Describe the steps in the program development process
* Create simple programs with input and output
* Use variables to store data
* Document a program using comments

**Unit 2: Writing programs using String objects and String class methods**

* Define string and understand its implementation in Java as A String class
* Input and output Strings
* Use simple String class methods to process String data
* Use String, char and int variable to store data related to strings
* Debug simple errors in programs

**Unit 3: Writing programs using primitive numeric data**

* Understand the difference between numeric data (int and double) and String objects
* Perform mathematical calculations with numeric data using mathematical operators following the order of operations
* Declare valid meaningful variables to reference data in a program
* Understand how to write/identify valid numeric data values
* Input and output numeric data
* Format numeric data for output
* Use methods of the Math class to perform numeric calculations

**Unit 4: Using conditional control structures to make decisions in a program**

* Introduce the if/else selection structures to make decisions
* Write Boolean expressions for conditions using reference operators and Boolean operators
* Introduce nested ifs to create multilevel selection structures
* Use String class methods to compare String objects

**Unit 5: Using loop control structure to repeat statements**

* Introduce the while loop and the for to repeat program statements
* Use a “counter” to control a loop
* Use loops to generate data to create a “chart”
* Discuss runtime errors associated with incorrectly constructed loops
* Use a loop to “traverse” a string character by character
* Create and use “nested” loop structures

**Unit 6: Using randomness in programs**

* Discuss the concept of randomness in a program
* Introduce the Random class to generate “pseudorandom” values in a program
* Create a “formula” to generate random int values in a specified range

**Unit 7: Methods**

* Discuss the concept of reusing code
* Create methods for repeated use
* Use methods to break down a difficult math idea
* Create appropriate documentation

**Unit 8: Civics**

* Discuss backing up computers
* Introduce the need for security
* Create password policies
* Discuss how to prevent online attacks
* Discuss file sharing legal issues
* Describe methods for continuing education

Portfolio Assignment:

To show that you have met all the programming expectation for Introduction to Computer Science, I am asking you to write the following program.

Write a program that will ask the user to select one of the 9 options from a menu, and perform the associated task. This should loop until the user enters option 9 (to quit). All user input must come in the main method, and be passed into the associated method, unless otherwise noted. All println methods should be done in the associated conditional statement (in the main method), unless otherwise noted.

If you have any questions, or need to meet to discuss questions, please email hcohen@ebnet.org

1. Make a menu that will only display the 9 choices below. It is a void method that will not accept input. It should display the 9 options. Your main method should ask the user to select a choice.

public class Menu(){  
 System.out.println(“Please select from the following:”);

System.out.println(“1 - …”)

…

}

1. Given 2 strings, a and b, return a string of the form short+long+short, with the shorter string on the outside and the longer string on the inside. The strings will not be the same length, but they may be empty (length 0). Hint: The length() method can be used to calculate the length of the word.

comboString("Hello", "hi") → "hiHellohi"

comboString("hi", "Hello") → "hiHellohi"

comboString("aaa", "b") → "baaab"

public String comboString(String a, String b) {

}

2) Given an int n, return true if it is within 10 of 100. Note: Math.abs(num) computes the absolute value of a number.

nearHundred(93) → true

nearHundred(90) → true

nearHundred(89) → false

public boolean nearHundred(int n) {

}

3) Given two int values, return their sum. Unless the two values are the same, then return double their sum.

sumDouble(1,2) -> 3

sumDouble(3,2) -> 5

sumDouble(2,2) -> 8

public int sumDouble(int a, int b) {

}

4) Given a positive integer n, calculate the sum of numbers from 1 to n. Note: You are NOT allowed to use the sum of numbers formula.

sumNumbers(5) -> 15 [1+2+3+4+5]

sumNumbers(4) -> 10 [1+2+3+4]

public int sumNumbers(int a){

}

5) Given a person’s height and weight, please calculate a person’s BMI.

BMI = weight \* 703 / height2

public double getBMI (double height, double weight){

}

6) Convert to Farenheit

Given a Celsius temperature, convert it to Fahrenheit

|  |
| --- |
| Fahrenheit(0) → 32  Fahrenheit(1) → 33.8  Fahrenheit(10) → 50 |

public double Fahrenheit(double a){

}

7) Use Random to flip a coin 20 times. Your method should print out “Heads” or “Tails.” Note: This is a void method.

public void flipCoin(){  
Random gen = new Random();

}

8) Given an integer, return the sum of the digits.

sumDigit(357) -> 15  
sumDigit(1001) -> 2

public int sumDigits(int a){

}

9) Quit the program, and thank them for playing.