

# C++ PROGRAMMING (335)

## REGIONAL 2020

**Production:**

Production Points \_\_\_\_\_ (300 points)

***TOTAL POINTS*** \_\_\_\_\_ (***300 points***)

**Failure to adhere to any of the following rules will result in disqualification:**

- 1. Contestant must hand in this test booklet and all printouts. Failure to do so will result in disqualification.**
- 2. No equipment, supplies, or materials other than those specified for this event are allowed in the testing area. No previous BPA tests and/or sample tests or facsimile (handwritten, photocopied, or keyed) are allowed in the testing area.**
- 3. Electronic devices will be monitored according to ACT standards.**

No more than sixty (60) minutes testing time

Property of Business Professionals of America.  
May be reproduced only for use in the Business Professionals of America  
*Workplace Skills Assessment Program* competition.

Write a program to display all the prime and perfect numbers up to a user specified number. Your program should use or do the following:

**Program output:**

1. The user should be asked to enter a number at the start of the program
2. Starting at 0 and going up to the entered value, any prime number is displayed, comma separated
3. Starting at 0 and going up to the entered value, any perfect number is displayed, comma separated
4. Each type of number should have its own clean screen to be displayed on.
5. The user should be asked to press a key to continue to the next type of number.
6. When finished ask the user if they would like to run the program again.

**Program requirements: - All of these should be used in a logical appropriate way.**

1. Use the modulo operator.
2. Include a for loop.
3. Include a do while loop.
4. Implement an isPerfect function that returns a bool and takes an integer.
5. Implement an isPrimefunction that returns a bool and takes an integer.

**Description of Number Types:**

Prime Numbers - A natural number greater than 1 that is divisible only by itself and 1. As such, it cannot be formed by multiplying any other two numbers together. Ex: 2, 3, 5, 7, 11, 13, 17, and 19.

Perfect Numbers - A natural number greater than 1 that is equal to the sum of its positive divisors, excluding itself. Ex: 6, 28, 496, and 8128.

**Calculation of Number Types:**

Calculation of Prime Numbers can be done using a for loop and modulo.

Calculation of Perfect Numbers can be done using either a for loop and modulo, or the equation  $(2^{p-1}) * 2^p$  where  $2^{p-1}$  is a prime number. For this test, you can use either method.

*Your application will be graded on the following criteria:*

**Solution and Project**

Project was found on the flash drive. \_\_\_\_\_ 10 points  
Project is named following the naming convention. \_\_\_\_\_ 10 points

**Program Execution**

Program runs. \_\_\_\_\_ 20 points

**If program does not execute, then remaining items in this section are not scored.**

At the start, user prompted to enter number to count up to. \_\_\_\_\_ 10 points  
The program labels and lists prime numbers. \_\_\_\_\_ 30 points  
The program labels and lists perfect numbers. \_\_\_\_\_ 40 points  
The program displays a clean screen between types. \_\_\_\_\_ 15 points  
The program pauses before clearing the screen. \_\_\_\_\_ 15 points  
The program asks if the user wants to run again. \_\_\_\_\_ 20 points

**Source Code Review**

Contestant ID is commented at the top of the program. \_\_\_\_\_ 5 points  
Code is commented at the top and as needed. \_\_\_\_\_ 5 points  
Code uses reasonable and consistent variable naming conventions. \_\_\_\_\_ 10 points  
Code uses an isPrime function to check prime numbers. \_\_\_\_\_ 30 points  
Code uses an isPerfect function to check perfect numbers. \_\_\_\_\_ 30 points  
Code uses modulo to calculate type in a logical and efficient way. \_\_\_\_\_ 20 points  
Code has function prototype declaration at the top of source file. \_\_\_\_\_ 10 points  
Code uses both loops in a logical and efficient way. \_\_\_\_\_ 20 points

**Total** \_\_\_\_\_/300 points