

CIS (050.3) Program Assessment: Teacher Guide

Spring 2018 | Community College of Allegheny County

This is a document for Java teachers administering the assessment instrument for the CCAC AS in Computer Information Science in CIT-111 during Spring 2018.

There is a companion document containing only the student handout linked here.

Our assessment focuses on the following program objective:

Program Objective 1: Apply core principles and practices of computing (Program completion objective #1)

Table of Contents

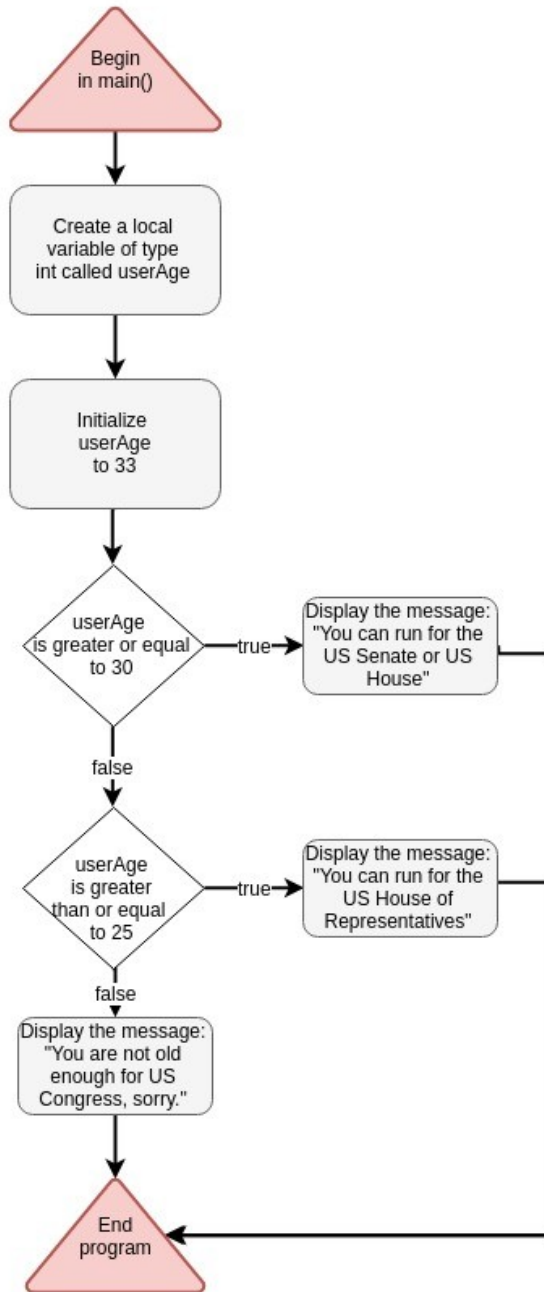
| | |
|---|---|
| Section 1: Student Instrument..... | 2 |
| Section 2: Scoring Rubric for task 1..... | 4 |
| Section 3: Completed code for task 1: Flow chart translation into Java..... | 5 |
| Section 4: Scoring rubric for task 2: Paper compiling a basic method call..... | 6 |
| Section 5: Sample student response for task 2: Paper compiling a basic method call (rubric score: 4)..... | 7 |

Section 1: Student Instrument

Task instructions:

This assessment measures your ability to

1. Comprehend a program flow chart and translate that information into a short application programmed in Java.
2. "Paper compile" a program written in Java to generate an expected program output. This program consists of a single class with two methods.



Task 1: Programming from a flow chart

1. Study the flow chart below and prepare to translate the flow into Java.
2. Create a class called OfficeElegibility and include a main() method.
3. Code the program logic shown in the flow chart inside your main() method.
4. When completed, test your code by changing the initial value of userAge.
5. Comment your code as needed.
6. Print your code onto paper and staple it to this assessment document.

Flow chart symbol Key:

Triangles represent program beginning and termination.

Rounded rectangles are general program events.

Diamonds represent decision points in the program, otherwise known as flow-of-control mechanisms.

Task 2: Paper compiling with method calls

Study the Java code below. This class called BasicMath carries out a few simple mathematical operations on two variables. Your job is to carefully track the value of each variable through the flow of the program and accurately generate the expected output of this program, which will consist of a single integer.

NOTE: After studying the code, please respond to the questions below. Showing any notes or scratch calculations will be useful for assessing your skill level. You are suggested to write your "best guess" about the value of each variable.

```
1  package essentials;
2
3  public class BasicMath {
4      public static void main(String[] args){
5          int a = 6;
6          int b = 4;
7
8          int result = doMath(b, a);
9
10         result = result + 1;
11
12         System.out.println(result);
13     }
14
15     public static int doMath(int x, int y){
16
17         return ((x + 2) * 2) + y;
18     }
19 }
20
```

1. What is the value of x at line 15?
2. What is the value of y at line 15?
3. What is the value of result AFTER line 8 has been executed?
4. What is the value of result when it is printed to the console on line 12?

Section 2: Scoring Rubric for task 1 – Flow Charts

Scoring rubric for task 1: Flow chart translation into Java

| Score | Criteria |
|--------------------------------|--|
| 0 - Not attempted | <ul style="list-style-type: none">• No Java code was attached to the assessment or the code was in no ways related to the flow chart |
| 1 - Insufficient skills | <ul style="list-style-type: none">• Code would NOT compile• A class with the appropriate name was created.• Decision logic either not attempted or the attempt did not relate to the specification |
| 2 - Below proficient | <ul style="list-style-type: none">• Code WOULD compile• Class and main() method were correctly created. Variables were correctly initialized.• Decision logic is attempted but does not match specification's requirements |
| 3 - Proficient | <ul style="list-style-type: none">• Code WOULD compile• Decision logic is correctly implemented |
| 4 - Advanced | <ul style="list-style-type: none">• Code WOULD compile• Decision logic is correctly implemented• Exhibits one or more of these features:<ul style="list-style-type: none">◦ Java code reveals thought aimed at eliminating duplicate code◦ Compactly implementing chained "if/else" statements◦ Comments reveal grasp of the flow of the code◦ Features of the code reveal thorough logic testing |

Section 3: Completed code for task 1: Flow chart translation into Java

```
1 package essentials;
2
3 public class OfficeEligibility {
4     public static void main(String[] args){
5         int userAge = 33;
6
7         if(userAge >= 30){
8             System.out.println("You can run for the US Senate or House");
9         } else if(userAge >= 25){
10            System.out.println("You can run for the US House of Representatives");
11        } else {
12            System.out.println("You are not old enough for US Congress, sorry");
13        } // close if-else blocks
14    } // close main
15 } // close class
16
```

Section 4: Scoring rubric for task 2: Paper compiling a basic method call

| Score | Criteria |
|--------------------------------|---|
| 0 - Not attempted | <ul style="list-style-type: none"> • 2 of the four questions were not answered. • If one or two were answered, they were incorrect |
| 1 - Insufficient skills | <ul style="list-style-type: none"> • The values for x and y were incorrectly identified (questions 1 and 2 are incorrect) • No evidence reveals that the student was tracing the logic in doMath() through notes or marks on the paper. |
| 2 - Below proficient | <ul style="list-style-type: none"> • The values for x and y at line 15 were incorrectly identified. • The logic inside doMath() was not properly applied and did not properly get returned to line 8 but an attempt was made to calculate the return value. • The final value of result at line 8 and 12 were incorrect. |
| 3 - Proficient | <ul style="list-style-type: none"> • The values for x and y at line 15 were correctly identified. • The logic within doMath() is correctly applied. • The incrementing of result was improperly applied • The program output was incorrect |
| 4 - Advanced | <ul style="list-style-type: none"> • a and b were properly passed to doMath • The math logic in doMath was properly executed • doMath() returned the correct result • The value of result was correct |

Section 5: Sample student response for task 2:

Paper compiling a basic method call (rubric score: 4)

Task 2: Paper compiling with method calls

Study the Java code below. This class called BasicMath carries out a few simple mathematical operations on two variables. Your job is to carefully track the value of each variable through the flow of the program and accurately generate the expected output of this program, which will consist of a single integer.

NOTE: After studying the code, please respond to the questions below. Showing any notes or scratch calculations will be useful for assessing your skill level. You are suggested to write your "best guess" about the value of all three of you boys..

```
1 package essentials;
2
3 public class BasicMath {
4     public static void main(String[] args){
5         int a = 6;
6         int b = 4;
7
8         int result = doMath(b, a);
9
10        result = result + 1;
11
12        System.out.println(result);
13    }
14
15    public static int doMath(int x, int y){
16
17        return ((x + 2) * 2) + y;
18    }
19 }
20
```

** Rubric score: 4*

Handwritten annotations:

- Line 8: $4 \ 6$ above `doMath(b, a)`; $18 \ 19 \ +1$ above `result = result + 1;`
- Line 17: $4 \ 6 \ 4$ above `((x + 2) * 2) + y;`; $(6^2) + 6$ below; $12 + 6$ below; 18 below.
- Arrows indicate flow: from `doMath(b, a)` to `result` (value 18), then to `result + 1` (value 19), and finally to `println`.

1. What is the value of x at line 15?

4

2. What is the value of y at line 15?

6

3. What is the value of result after line 8 has been executed?

18

4. What is the value of result when it is printed to the console on line 12?

19