

AP Computer Science Practice Problems

1. What is printed for each of the print statements below? Draw the console output in the correct format with the correct values. [4 points]

```
int n = 3;
double result;

result = (n + 1) * n / 2;
System.out.println(result);
result = (n / 2) * (n + 1);
System.out.println(result);
result = (1 / 2) * n * (n + 1);
System.out.println(result);
```

6.0
0.0
0.0

$(n+1) * n / 2$
 $(3+1) * 3 / 2$
 $4 * 3 / 2$
 $12 / 2$
 $(n/2) * (n+1)$
 $(3/2) * (3+1)$
 $0 * 4$

line 2 = 0

$result = (n+1) * n / 2$
 $(3+1) * 3 / 2$
 $4 * 3 / 2$
 $12 / 2$

line 1 = 6

$(1/2) * 3 * (3+1)$
 $0 * 3 * 4$

line 3 = 0

2. Draw the console output for the code below in the correct format with the correct values. [2 points]

```
int a = 1, b = 2;
double ratio;
ratio = (double)a / b;
System.out.println(ratio);
ratio = (double)(a / b);
System.out.println(ratio);
```

$ratio = (double)(1/2)$

$ratio = (double)(0)$

0.0

$ratio = (double)a / b$

$ratio = (double) 1 / 2$

$ratio = 1.0 / 2$

$ratio = 1.0 / 2.0$ ← autocast

ratio = 0.5

0.50.0

3. Which of the following expressions does **NOT** evaluate x to 0.4? [1 point]

- A. `double x = (int) 4.5 / (double) 10;`
- B. `double x = (double) (4 / 10);`
- C. `double x = 4.0 / 10;`
- D. `double x = 4 / 10.0;`
- E. `double x = (double) 4 / (double) 10;`

← int div first, then cast

4. What is printed below? Draw the correct console output in the correct format with the correct values. [3 points]

```
int r = 0;
r = 17 % 3;    = 2
System.out.print(r); ←
r = 8 % 2;     = 0
System.out.print(r); ←
r = 4 % 5;     = 4
System.out.print(r); ←
```

no newline

5. What is the result when the code segment is compiled and executed? [1 point]

```
int m = 4, n = 5;
double d = Math.sqrt((m + n) / 2);
System.out.println(d);
```

$\Rightarrow \text{Math.sqrt}((4+5)/2)$
 $\text{Math.sqrt}(9/2)$
 $\text{Math.sqrt}(4.5)$
 $\Rightarrow 2.0$

- A. Syntax error "Math.sqrt(double) cannot apply to int"
- B. 1.5 is displayed
- C. 2.0 is displayed
- D. 2.1213203435596424 is displayed
- E. ClassCastException

6. Given the declarations and code below which of the following expressions will evaluate to 7.5? *decided* [1 point]

`int p = 5, q = 3;`

- 5.0* *also cast to double: 2.0*
- 1 ~~A.~~ `(double) p * (double) q / 2;`
 - 2 ~~B.~~ `(double) p * (double)(q / 2);` *int div*
 - 3 ~~C.~~ `(double) (p * q / 2);` *int mult and div*

- A. I only
 B. II only
 C. I and II
 D. I, II, and III
 E. None of the above

7. Assuming that *c* and *d* are Boolean variables,

`boolean c = <some value>;`
`boolean d = <some value>;`

the expression `!c || d` is equivalent to

[1 point]

- A. `!(c && d)`
 B. `!(c && !d)`
 C. `c && !d`
 D. `!(c || !d)`
 E. `!(!c && d)`

!c || d
!(c && !d)

factor

8. Suppose a , b , and c are positive integers under 1000 and x satisfies the formula after the declarations

```
int a, b, c, d;
```

```
a / b = c / x;
```

$$\frac{a}{b} = \frac{c}{x} \leftarrow d$$

The integer value d is obtained by truncating x to an integer. Which of the following code segments correctly calculates d ? [1 point]

- A. $d = c * b / a;$
- B. `int temp = c * b;`
 $d = \text{temp} / a;$
- C. `(int) temp = b / a;`
 $d = c * \text{temp};$

$$\frac{a}{b} = \frac{c}{\frac{x}{1}}$$

$$bc = a \left(\frac{x}{1} \right)$$

$$\frac{bc}{a} = \left(\frac{x}{1} \right)$$

- A. I only
- B. II only
- C. I and II
- D. II and III
- E. I, II, and III

9. Suppose the class **Particle** has the variables defined:

[1 point]

```
public class Particle
{
    public static final int START_POS = 100;
    private double velocity;

    /*other code not shown*/
}
```

Which of the following inside the Particle class is true?

- A. `velocity` can be passed as an argument to a method, but `START_POS` cannot.
- B. Java syntax rules wouldn't allow us to use the name `startPos` instead of `START_POS`.
- C. Statement `double pos = START_POS + velocity` is a syntax error.
- D. Java syntax rules wouldn't allow us to make `velocity` public.
- E. Statement `double START_POS += velocity` is a syntax error.

can't be changed

10. What is the output from the following code segment?

[1 point]

```
double pi = 3.14159;  
int r = 100;  
int area = (int)(pi * Math.pow(r, 2));  
System.out.println(area);
```

Handwritten notes:
- 100^2 above `Math.pow(r, 2)`
- *4 zeroes* above `(int)`
- $(int)(\pi * 10,000)$
- $(int)(31415.9)$

- A. 30000
- B. 31415
- C. 31416
- D. 314159
- E. It depends on the particular computer system.

11. What is the output from

[1 point]

```
int n = 12;  
System.out.print(goFigure(n));  
System.out.print(" " + n);
```

```
public static double goFigure(int n)  
{  
    n = n % 7;  
    return (double)(12 / n);  
}
```

Handwritten notes:
- *no new line* with an arrow pointing to the first `System.out.print`
- *dummy* with an arrow pointing to `(12 / n)`

- A. 2.4 12
- B. 2.4 6
- C. 2.4 5
- D. 2.0 12
- E. 2.0 5

Handwritten calculation:
 $n = 12 \% 7 = 5$
 $= \text{return}(\text{double})(12 / 5)$
 $\text{double}(2.4)$
 $= \text{return} 2.0$

12. What is the size of a **double** variable in Java?

[1 point]

- A. 2 bytes
- B. 4 bytes
- C. 8 bytes
- D. It depends on the compiler settings.
- E. It depends on the operating system.

13. What does the print statement below display? *↑ autocast to String* [1 point]

```
System.out.println("1" + new Integer(2) + 3);
```

- A. The statement has a syntax error and won't compile.
- B. 6
- C. 15
- D. 123
- E. ClassCastException

14. Given [1 point]

```
double x = 5, y = 2;
```

What is the value of *m* after the following statement is executed?

```
int m = (int)(x + y + x / y - x * y - x / (10 * y));
```

- A. -1
- B. -0.75
- C. -0.5
- D. 0
- E. 1

15. What is the output from the following code? [1 point]

```
int a = 1, b = 2, c = 3;
```

```
a += b + c;
```

```
b += a + c;
```

```
c += a + b;
```

```
System.out.println(a + " " + b + " " + c);
```

*1 += 2+3
2 += 6+3
3 += 6+11*

*1 += 5
2 += 9
3 += 17*

*a = 6
b = 11
c = 20*

- A. 3 3 4
- B. 3 5 6
- C. 5 4 3
- D. 5 8 13
- E. 6 11 20

16. What is the size of an `int` variable in Java?

[1 point]

- A. 2 bytes
- B. 4 bytes
- C. 8 bytes
- D. It depends on the compiler settings.
- E. It depends on the operating system.

17. The method `xProperty` is defined as follows:

[1 point]

```
public boolean xProperty(int a)
{
    return a == 2 * (a / 10 + a % 10);
}
```

For which of the following values of `a` does `xProperty` return `true`?

- A. 2
- B. 4
- C. 18
- D. 28
- E. 128

18. Which of the following does NOT display 2/3?

[1 point]

- A. `System.out.println("2/3");`
- B. `System.out.println("2" + "/" + "3");`
- C. `System.out.println(2/3);`
- D. `System.out.println(2 + "/" + 3);`
- E. `System.out.println((int)2 + "/" + (int)3);`

19. What is the size of a **Boolean** variable in Java?

[1 point]

- A. 1 bit
- B. 2 bits
- C. 2 bytes
- D. 4 bytes
- E. 8 bytes

20. Draw the console output from the following code.

[3 points]

```
System.out.println("Hello");  
System.out.print("World");  
System.out.println("Look at me!");
```

Handwritten output:
Hello (newline)
WorldLook at me!

no new
line

newline

21. Consider the following method:

[1 point]

```
public static int compute(int val)  
{  
    val += val;  
    val += val;  
    return val;  
}
```

6

4

4 + = 4

8 + = 8

return 16;

6 + = 6

12 + = 12

return 24;

Which statement could replace the body of compute so that the same result is returned?

- A. **return** (int)Math.pow(val, 4);
- B. **return** (int)Math.pow(val, 3);
- C. **return** 2 * Math.pow(val, 2);
- D. **return** val * 4;
- E. None of the above

= 32

4 * 4 = 16

6 * 4 = 24

Don't need to
instantiate to

22. The method below is part of a class called **Computer**. Which is the correct way to invoke the method `compute` outside of the class **Computer**? [1 point]

```
public static int compute(int val)
{
    val += val;
    val += val;
    return val;
}
```

- A. `Math.compute(3);` ← wrong module
B. `Computer c = new Computer(); c.compute(3);`
C. `Computer.compute(3);`
D. `Computer.compute(3.0);`
E. `Computer.compute("3");`

integer

23. What is returned from a call to the method below if the value passed to the method is 5? [1 point]

```
public static int compute(int val) // 5
{
    val += val;    5 += 5
    val += val;    10 += 10
    return val;    20
}
```

- A. 125
B. 25
C. 20
D. 100
E. 625

24. Consider the following code:

[1 point]

```
public static final int START = 100;
```

Which of the following is not legal code?

- A. `int max = START + 200;`
- B. `int min = START - 100;`
- C. `System.out.println("Start Squared = " + Math.pow(START, 2));`
- D. `value += 5 + START; //assume value was created and initialized`
- E. `START += START + 20;`

25. Suppose that the partial Desk class is written as follows:

[1 point]

```
public class Desk
{
    private String typeOfMaterial;

    public Desk(String type)
    { /*implementation not shown*/ }
}
```

Which code is the correct code to instantiate a Desk object?

- I. `Desk desk = new Desk();`
- II. `Desk desk = new Desk("wood");`
- III. `Desk desk = Desk.wood;`

- A. I only
- B. II only
- C. I and II only
- D. I, II , and III
- E. II and III only

26. Which method below is the correct accessor (getter) method implementation that can be placed in the Desk class? [2 points]

- A. **private** String getTypeOfMaterial(){ **return** typeOfMaterial;}
- B. **private** Desk getTypeOfMaterial(){ **return** typeOfMaterial;}
- C. **public** Desk getTypeOfMaterial(){ **return** typeOfMaterial;}
- D. **public** String getTypeOfMaterial(String t){ typeOfMaterial=t;}
- E. **public** String getTypeOfMaterial(){ **return** typeOfMaterial;}

27. The Desk class is used to construct a desk. Write the code that will construct a desk made of wood and write the setter for typeOfMaterial. [2 points]

```
public class Desk
{
    private String typeOfMaterial;
    public Desk(String type)
    { /*implementation not shown*/ }
}
```

28. Suppose you were tasked with computing the minimum of three integer scores: a, b, c. Which method call performs this calculation? [1 point]

- I. Math.min(Math.min(a, b), c);
- II. Math.min(a, b, c);
- III. Math.min((a, b), c);

- A. I only
- B. II only
- C. I and II
- D. I and III
- E. I, II , and III

29. Consider the incomplete class definition and the code that is written in a client program: [1 point]

```
public class Mail{ ... }  
Mail mail = new Mail();
```

Which statement is **NOT** correct?

- A. mail is an object.
- B. Mail is the blueprint for mail.
- C. mail is an instance of Mail.
- D. A Mail is an instance of mail.
- E. Mail is a class.

other way

30. Which is the line of code that will print 64.0? [1 point]

- A. System.out.println(Math.pow(2,4)); = 16
- ~~B.~~ System.out.println(Math.sqrt(64)); = 8
- ☒ C. System.out.println(Math.pow(4,3)); = 64.0
- ~~D.~~ System.out.println(Math.sqrt(8)); = $\sqrt{8}$
- ~~E.~~ System.out.println(Math.pow(3, 2)); = 9

31. What is the output from [1 point]

```
double n = 12;  
System.out.print(n);  
n = goFigure(n);  
System.out.print(" " + n);
```

```
public static double goFigure(double n)  
{  
    n = (int)n % 7;  
    return 12 / n;  
}
```

- A. 12 2.0
- B. 12.0 2.0
- C. 12 2.4
- ☒ D. 12.0 2.4
- E. 12.0 5.0

$12 \% 7 \rightarrow n = 5.0$ - autocast double
 $12 / 5.0 \rightarrow 2.4$ - auto cast double

32. The math class has a method called random. It generates a random number on the interval [0, 1). There is also a Random object that can produce a random set of integers. This random object is created using its default constructor.

a. What random integer interval will the following code produce? **[2 points]**

```
(int)(Math.random() * 5) + 10;
```

b. What random integer interval will the following code produce? **[2 points]**

```
Random rand = new Random();  
rand.nextInt(13) - 6 ;
```

33. What is the output from the following code?

[1 point]

```
int a = 1, b = 2, c = 3;  
a += b + c;  
b += a + c;  
c += a + b;  
System.out.println(a + " " + b + " " + c);
```

- A. 3 3 4
- B. 3 5 6
- C. 5 4 3
- D. 5 8 13
- E. 6 11 20

34. Which of the following statement is correct?

[1 point]

```
public class Square extends Rectangle {}
```

```
public class Rectangle extends Quadrilateral {}
```

```
public class Quadrilateral {}
```

- I. A Rectangle is an Object
- II. Square is a subclass of Rectangle
- III. Quadrilateral is the super class of Rectangle

- A. I & II Only
- B. II & III Only
- C. II Only
- D. I & III Only
- E. I, II, & III



35. The Pencil class below has two constructors. Which of the following are valid instantiation of an object from the Pencil class?

[1 point]

```
public class Pencil {
```

```
//attributes not shown
```

```
Pencil(String type, boolean hasEraser) { //implementation not shown}
```

```
Pencil(String type, boolean hasEraser, int number) { //implementation not shown}
```

```
//methods not shown }
```

- I. new Pencil("Mechanical", true, 7/3);
- II. String type = "Regular";

```
new Pencil(type, false, 1);
```

- III. new Pencil("Regular", true);

- A. I only
- B. II only
- C. I and II
- D. II and III
- E. I, II , and III

Handwritten note: *int div* with an arrow pointing to the `7/3` in option I.

36. Assuming that *c* and *d* are Boolean variables,

[1 point]

```
boolean c = <some value>;
```

```
boolean d = <some value>;
```

the expression `!c || d` is equivalent to

- A. `!(c && d)`
- ☒ B. `!(c && !d)`
- C. `c && !d`
- D. `!(c || !d)`
- E. `!(!c && d)`

Handwritten notes:
 $\overline{c \ \&\&\ \overline{d}}$
 $\overline{c} \ ||\ d$

37. Which of the following statement is correct?

[1 point]

- A. All attributes are methods. *variables*
- ☒ B. All classes must have a non-default constructor.
- C. All instance variables are private. *no, public too*
- D. All methods must return a value. *void*
- E. All final variables are constants.

object has that

38. Which one stands for API?

[1 point]

- A. Application Portable Information
- B. Application Portable Invariant
- C. Abstract Portable Information
- D. Abstract Programming Interface
- E. Application Programming Interface

39. The String class has a method called **contains**. It returns true if a String contains another String. For example, "candy".contains("a") returns true and "candy".contains("e") returns false. The method below should use contains to count the number of **different** vowels that are in a String. For example, the String "candy" should return 1 because there is an "a". However, the String "beautiful" should return 4 because it contains four **different** vowels "eoui". The maximum number this method can return is 5 because there are exactly five vowels "aeiou". If the String does not contain a vowel, -1 is returned. For example, the String "gym" will return -1 because y is not a vowel. [6 points]

//Precondition: s is not null & contains only lowercased letters

//Postcondition: returns vowel count, otherwise returns -1

public int countDifferentVowelsInString(String s)

```
{
    int number = 0;
    String vowels = "aeiou";
    for (char letter : vowels)
        number += s.contains(String.valueOf(letter));
    if (number == 0)
        return -1;
    return number;
}
```

↑
using int conversion from
boolean



40. The class `Palindrome` has one constructor and two methods. One of the methods is called **`reverseString`** that returns a `String` with the characters in reversed order. The implementation of **`reverseString`** is not shown but it works as intended. Write the constructor and method called **`isPalindrome`** that takes in one `String` and returns a `boolean`. The method's job is to determine if the word is a palindrome. A palindrome is a `String` that spells the same word forwards and backwards. For example, "dad", "mom", "racecar", "pop" are all palindromes. The `String` class has a method called **`equals`** that returns true if two words have the same characters in the same order. For example, "mom".equals("mom") returns true and "candy".equals("ydnac") returns false. [6 points]

```
public class Palindrome()
{
    private String word;

    //Precondition: s in NOT null
    //Postcondition: returns s with characters reversed
    private String reverseString(String s)
    { /*Implementation not shown*/ }

    //Complete the constructor [2 points]
    public Palindrome(String word)
    {
        this.word = word;
    }

    //Complete the method [4 points]
    //Precondition: s in NOT null
    //Postcondition: return true if word is a palindrome
    //else returns false otherwise
    public boolean isPalindrome()
    {
        return this.word.equals( this.reverseString(this.word) );
    }

    //end class Palindrome
```