

Part I. (55 points) Solve each of the following problems. For the multiple choice problems, select the correct answer by placing an “X” in the box beside it.

- (1^{pt}) 1. Which of the following choices is a legal and legitimate **Java** variable name?

☐ 2bad4you
☐ calvin&hobbes
☐ year2000
☐ #hammertime

1 pt

- (1^{pt}) 2. You would like to set up a variable called **ounces** that has the value 16. What simple **Java** statement will accomplish this?

☐ `int ounces = 16;`
☐ `int 16 = ounces;`
☐ `public static int ounces(16)`
☐ `ounces(16);`

1 pt

- (1^{pt}) 3. What is the output of the following **Java** code?

`System.out.println(19 % 5);`

☐ 3
☐ 0
☐ 4
☐ 1

1 pt

- (1^{pt}) 4. What is the output of the following **Java** code?

`System.out.println(1 / 3);`

☐ 0.3333333333333333
☐ 0
☐ 0.3
☐ It will give a compile-time error.

1 pt

- (1^{pt}) 5. What is the correct data type for decimal numbers such as 3.14159?

☐ `double`
☐ `int`
☐ `boolean`
☐ `String`

1 pt

- (1^{pt}) 6. What is the correct data type for text data such as "hello world"?

☐ `double`
☐ `int`
☐ `boolean`
☐ `String`

1 pt

6 pts

- (1^{pt}) 7. What is the value of `amount` after executing the following Java code?

```
String dinner = "Hamburger";  
int amount = dinner.length();
```

- ☐ 8
☐ 9
☐ 10
☐ 11

1 pt

- (1^{pt}) 8. What is the value of `position` after executing the following Java code?

```
String lunch = "Pizza";  
int position = lunch.indexOf("z");
```

- ☐ 0
☐ 1
☐ 2
☐ 3

1 pt

- (1^{pt}) 9. What is the value of `first` after executing the following Java code?

```
String breakfast = "Pancakes";  
String first = breakfast.substring(0, 1);
```

- ☐ P
☐ Pan
☐ cakes
☐ Pancakes

1 pt

- (1^{pt}) 10. Which of the following choices is a Java reserved keyword?

- ☐ console
☐ while
☐ memory
☐ result

1 pt

- (1^{pt}) 11. Which of the following is a TRUE statement about the `String` data type?

- ☐ `String` is a primitive data type.
☐ The standard Java library has a predefined class called `String`.
☐ `Strings` can only contain numbers and digits, not punctuation.
☐ `Strings` are mutable, once they are created they can be changed or altered.

1 pt

- (1^{pt}) 12. What is the data type of the following variable: `num = 42;`

- ☐ boolean
☐ double
☐ `String`
☐ `int`

1 pt

6 pts

(1^{pt}) 13. What is the output of the following Java code?

```
for (int i = 3; i <= 12; i++)  
{  
    System.out.print(i + " ");  
}
```

- ☐ 5 6 7 8 9
☐ 4 5 6 7 8 9 10 11 12
☐ 3 5 7 9 11
☐ 3 4 5 6 7 8 9 10 11 12

1 pt

(1^{pt}) 14. What is the output of the following Java code?

```
String greetings = "Hello World!";  
System.out.println(greetings.substring(6));
```

- ☐ Hello World!
☐ The Java code will not compile.
☐ World!
☐ Hello

1 pt

(1^{pt}) 15. What is the output of the following Java code?

```
String weather = "One Fine Day";  
String result = weather.substring(4, 8);  
System.out.println(result);
```

- ☐ Fine
☐ One Fine
☐ Fine Day
☐ Day

1 pt

(1^{pt}) 16. Which of the following choices demonstrates the correct way to concatenate two Strings together?

- ☐ String answer = "Good" == "Burger";
☐ String outcome = "Best" + "Pizza";
☐ String display = "Fresh" <> "Salad";
☐ String result = "Ripe" / "Fruit";

1 pt

(1^{pt}) 17. Consider the following Java code:

```
String drink = "sprite";  
String beverage = "pepsi";
```

How would you determine if these two Strings are the same, or different?

- ☐ Divide one String by the other. If the result is one, then the Strings are equal.
☐ Use the differential() method in the following manner:
 double outcome = drink.differential(beverage);
☐ Use the equals() method in the following manner:
 boolean result = drink.equals(beverage);
☐ Use the == operator in the following manner:
 boolean answer = (drink == beverage);

1 pt

5 pts

(1^{pt}) **18.** Which of Java's primitive data types would be most suitable to store the square root of 2?

1 pt

(1^{pt}) **19.** Which of Java's primitive data types would be most suitable to store your age?

1 pt

(1^{pt}) **20.** Write a single line of code that will create an integer variable called `num` and store the number 407 in it.

1 pt

(1^{pt}) **21.** Write a single line of code that will increment the previously declared integer variable `num` by 1.

1 pt

(1^{pt}) **22.** What are the two possible values of a `boolean` variable?

1 pt

(1^{pt}) **23.** What is the Java operator for the boolean AND operation?

1 pt

(1^{pt}) **24.** What is the Java operator for the boolean OR operation?

1 pt

(1^{pt}) **25.** Write a single line of code that will create a `String` variable called `name` and store the text "Bob" in it.

1 pt

(1^{pt}) **26.** When comparing two Strings for equality, the assignment operator(==) should not be used. What is the name of the method that *should* be used?

1 pt

(1^{pt}) **27.** Write code using a `for` loop that will print out the numbers: 2 4 6 8 10

1 pt

10 pts

(2^{pts}) **28.** Convert the following binary(base-2) numbers to decimal(base-10).

(a) (1 pt) 1011

(b) (1 pt) 10001

2 pts

(2^{pts}) **29.** Convert the following hexadecimal(base-16) numbers to decimal(base-10).

(a) (1 pt) A7

(b) (1 pt) 2E

2 pts

(2^{pts}) **30.** Convert the following binary(base-2) numbers to hexadecimal(base-16).

(a) (1 pt) 10010011

(b) (1 pt) 110010100001

2 pts

(2^{pts}) **31.** Convert the following hexadecimal(base-16) numbers to binary(base-2).

(a) (1 pt) B4

(b) (1 pt) 9C

2 pts

8 pts

(1^{pt}) **32.** What is the output of the following **while** loop?

```
int num = 0;
while (num < 3)
{
    System.out.println(num);
    num++;
}
```

1 pt

(1^{pt}) **33.** What is the output of the following **while** loop?

```
int num = 5;
while (num < 10)
{
    System.out.println(num);
    num += 2;
}
```

1 pt

(1^{pt}) **34.** What is the output of the following **for** loop?

```
for (int i = 0; i <= 8; i += 2)
{
    System.out.println(i);
}
```

1 pt

(1^{pt}) **35.** What is the output of the following **for** loop?

```
for (int i = 5; i >= 1; i--)
{
    System.out.println(i);
}
```

1 pt

4 pts

- (2^{pts}) **36.** Write a Java function that converts a Fahrenheit temperature to a Celsius temperature using the following equation:

2 pts

$$\text{Celsius} = \frac{5.0}{9.0} * (\text{Fahrenheit} - 32.0)$$

Your function should be called `public static double temperature(double fahrenheit)`, which takes in a single parameter, `fahrenheit`. The function should return a `double` which is the Celsius conversion of `fahrenheit`.

- The output of your program should be 30.0 if the following statements are executed:

```
double result = temperature(86.0);  
System.out.println(result);
```

- (2^{pts}) **37.** Write a Java function that computes the length of the hypotenuse of a triangle by using the Pythagorean theorem:

2 pts

$$c = \sqrt{a^2 + b^2}$$

Your function should be called `public static double pythagoras(double a, double b)`, which takes in two parameters, `a` and `b`. The function should return a `double` which is the length of the hypotenuse of the triangle. *Hint:* You may need to use the `Math` functions `pow()` and `sqrt()`.

- The output of your program should be 5.4671747731346585 if the following statements are executed:

```
double result = pythagoras(3.5, 4.2);  
System.out.println(result);
```

4 pts

- (4pts) **38.** In this question, you will write a Java function that generates the Hailstone sequence of integers. This is a sequence of numbers that goes up and down repeatedly, but eventually the sequence comes down to end in one. The algorithm for the Hailstone sequence can be expressed as follows:

4 pts

- Pick some positive integer and call it `num`.
- If `num` is even, divide it by two.
- If `num` is odd, multiply it by three and add one.
- Print out this new value of `num`.
- Continue this process until `num` is equal to one.

Write a Java function that takes in a starting value `num` as a parameter, and generates the Hailstone sequence from that value. Your function should be called:

```
public static void hailstone(int num)
```

Note that this function does not return a value. Instead, you must use: `System.out.println()` to display your values as they are being generated.

- If the following statement is executed:

```
hailstone(5);
```

Then the output of your program should be:

```
16
8
4
2
1
```

4 pts

- (4^{pts}) **39.** In this question, you will write a **Java** function that performs the multiplication operation, but with a technique that the Ancient Egyptians used. The algorithm for Ancient Egyptian Multiplication can be expressed as follows. Assume that **grow** and **shrink** are the numbers to be multiplied together:

4 pts

- Create an integer variable called **product** to hold the solution.
- Check to see if **shrink** is an odd number.
- If **shrink** is odd, then add the number **grow** to the variable **product**.
- Multiply the number **grow** by 2.
- Divide the number **shrink** by 2(*Note: Use integer division*).
- Continue until the number **shrink** becomes zero.

Write a **Java** function that takes in two integer values, **grow** and **shrink**, as parameters, and calculates their multiplicative product using the Ancient Egyptian Multiplication algorithm. Your function should be called:

```
public static int multiply(int grow, int shrink)
```

Note that this function returns an integer value.

- If the following statements are executed:

```
int result = multiply(23, 58);  
System.out.println(result);
```

Then the output of your program should be: 1334

4 pts

- (4^{pts}) **40.** Pig Latin is a type of slang language that is easy to learn and understand. An English word can be translated into Pig Latin by following these two simple rules:

- If the English word begins with a vowel, then the corresponding Pig Latin word is generated by appending the letters "hay" to the end of the word. For example, "orange" becomes "orangehay".
- If the English word begins with a consonant, then the corresponding Pig Latin word is generated by moving the first letter to the end of the word, then appending the letters "ay". For example, "peach" becomes "eachpay".

Write a Java function that takes in an English word as a parameter, and translates that word to Pig Latin. Your function should be called `public static String pigLatin(String word)`, which takes in a single parameter, `word`. The function should return a String which is the Pig Latin translation of `word`.

- The output of your program should be `orangehay` if the following statements are executed:

```
String result = pigLatin("orange");  
System.out.println(result);
```

- The output of your program should be `eachpay` if the following statements are executed:

```
String result = pigLatin("peach");  
System.out.println(result);
```

4 pts

4 pts