1. State clearly whether the following statements are true or false:

1. A class can have more than one constructor. True
2. The words “public” and “class” are part of the Java language. True
3. We always define the fields to be public. False
4. The lifetime of a variable defines the section of source code from which the variable can be accessed. True
5. The lifetime of a field is the same as the lifetime of the object to which it belongs. True
6. The method body is the remainder of the method after the header. True
7. A method returning a value means that some information is passed internally between two different parts of the program. True
8. Java does allow void methods to contain a special form of return statement in which there is no return value like: *return;* True
9. You can have a same name for a local variable as a field. False

2. Discuss the assignment statement and the conditional statement. Give an example of each.

The assignment statement sets a variable equal to a value. An example is:

number = 3

The conditional statement is an if-then statement that will take one of two possible actions based on the result of a test (which is defined after the if statement). An example is:

if (number > 0) {

total = number + total;

}

else {

System.out.println(“Please enter a number greater than 0.”);

}

3. What are the two features of the constructor that make it look significantly different from the

methods of the class?

The constructor has the same name as the class name, and they do not have any return type. The method can be named anything and must always have a return type.

4. What are the accessor and mutator methods? Give one example of each method.

Accessor methods return information (a value) about the state of an object. They usually use return statements to do so. An example is:

private getPrice()

{

return price;

}

Mutator methods change the state of an object, meaning they change the value of one or more fields. An example is:

private setPrice(int cost)

{

price = cost;

}

5. What is the difference between a global variable and a local variable? Explain.

A global variable can be accessed in any of the source code within the class it is declared in, while a local variable is declared and initialized within a method and can only be accessed within that method. The lifetime of a local variable is the same as the lifetime of the method. The lifetime of a global variable is the same as the lifetime of the program.

6. Write an **algorithm** to determine whether a given number is a prime.

private boolean isPrime(int number)

{

int i;

for (i == 2; i <= number; i++) {

if (n % i == 0) {

return false

}

}

return true;

}

7. Java is considered a strongly typed language. What does that mean?

Java is a strongly typed language because every variable must be declared with a data type.

8. What is the difference between a "compiler" and an "interpreter"?

A compiler scans the entire program and translates it as a whole into machine code. A Java compiler compiles source files (.java) to bytecode files (.class). An interpreter, on the other hand, translates programs one statement at a time. A java interpreter is usually referred to as the Java Virtual Machine (JVM), and it reads and executes the bytecodes in the .class files.

9. Explain the difference between *high-level languages* and *machine language.*

High-level language is a programming language that uses English and mathematical symbols and is therefore usually used by programmers to write programs. High-level language is not understood by the computer therefore it does need to be translated. On the other hand, machine language is the only language that is understood by the computer and therefore it doesn’t need to be translated.

10. What is a *variable?* Explain.

A variable is a piece of memory that can contain a data value.

OOP Design:

11. You task is to create a roster for our OOP class. Design a class to achieve this task. List the name of instance fields, name/signature of the methods, and the name of the class. How many mutator/accessor methods are appropriate to produce the usual reports for students? Think of how you can use this class to access/query different information about a student. Write out your design clearly below.

public class Roster {

private String firstName;

private String lastName;

private int idNumber;

private String attendance;

public Roster(String firstName, String lastName, int idNumber, String attendance)

{

this.firstName = firstName;

this.lastName = lastName;

this.idNumber = idNumber;

this.attendance = attendance;

}

public void printInformation()

{

System.out.println(“First Name: ” + firstName);

System.out.println(“Last Name: ” + lastName);

System.out.println(“idNumber: ” + idNumber);

System.out.println(“Attendance: ” + attendance);

}

public String getName()

{

System.out.println(“Name of Student: ” + firstName + “ ” + lastName);

}

public int getIDNumber()

{

System.out.println(“ID Number: ” + idNumber);

}

public String getAttendance()

{

System.out.println(“Attendance: ” + attendance)

}

public void setAttendance(String attendance)

{

this.attendance = attendance;

}

12. Write out a constructor for a class called Module. The constructor should take a single

parameter of type String called moduleCode. The body of the constructor should assign the

value of its parameter to a field called code. You don’t have to include the definition for code,

just the text of the constructor.

public Module(String moduleCode)

{

code = moduleCode;

}

13a. Understand the code for the following method:

**public String getLoginName()**

**{**

**return name.substring(0,4) +id.substring(0,3);**

**}**

What would be returned by getLoginName for a student with name "Henry Moore" and

id "557214"?

Henr557

b. Modify the getLoginName method of so that it always generates a login name, even if either

the name or the id field is not strictly long enough. For strings shorter than the required length,

use the whole string.

**public String getLoginName()**

**{**

**if (name.length() < 4 & id.length() < 3)**

**{**

**return name + id**

**}**

**else if (name.length() < 4)**

**{**

**return name + id.substring(0,3);**

**}**

**else if (id.length() < 3)**

**{**

**return name.substring(0,4) +id;**

**}**

**}**

14. Consider the following expressions. Try to predict their results, and then type them in the

Code Pad to check your answers (write your answers)

99 + 3 102

"cat" + "fish" “catfish”

"cat" + 9 “cat9”

9 + 3 + "cat" “12cat”

"cat" + 3 + 9 “cat39”

"catfish".substring(3,4) “f”

"catfish".substring(3,8) runtime error (throws a StringIndexOutOfBoundsException)

Did you learn anything you did not expect from the exercise? If so, what was it?

I learned that when a string and an int are added, they just become one string. I also learned that if the substring end index is greater than the number of characters in a string, you will have a runtime error and Java will throw a StringIndexOutOfBoundsException.

**Coding:**

15. Write a method to calculate the revenue from a sale based on the unit price and quantity of a

product input by the user (you can decide the name of the method, parameters, if any). The

discount rate is 10% for the quantity purchased between 100 and 120 units, and 15% for the

quantity purchased greater than 120 units. If the quantity purchased is less than 100 units, the

discount rate is 0%. This method will print the number of units sold and the total purchase

price for the entire purchase, using appropriate label.

*(you can create a test class and run this method to make sure it works before submitting your code)*

private void revenue(double unitPrice, int quantity) {

if (quantity >= 100 & quantity <= 120) {

unitPrice = 0.9 \* unitPrice

}

else if (quantity > 120) {

unitPrice = 0.85 \* unitPrice

}

else {

unitPrice = unitPrice

}

totalPrice = unitPrice \* quantity

System.out.println(“Number of units sold: ” + quantity)

System.out.println(“Total purchase price: ” + totalPrice)

}