Students may use their text book, any hand written notes, any digital notes or programs written by the student (You may use your flash drive and internet). 8:00am – 11:00am

**SHORT ANSWER SECTION**

1. (10 Points) Explain the difference between Procedural Programming and Object Orientated Programming.

**Procedural programming** – It uses a list of instructions to tell the computer what to do step-by-step. Procedural programming relies on - you guessed it - procedures, also known as routines or subroutines. A procedure contains a series of computational steps to be carried out. Procedural programming is also referred to as imperative programming. Procedural programming languages are also known as top-down languages.

**Object-oriented programming**, or **OOP -** It is an approach to problem-solving where all computations are carried out using objects. An **object** is a component of a program that knows how to perform certain actions and how to interact with other elements of the program. Objects are the basic units of object-oriented programming. A simple example of an object would be a person. Logically, you would expect a person to have a name. This would be considered a property of the person. You would also expect a person to be able to do something, such as walking. This would be considered a method of the person.

1. (5 points) How are objects created in Java? List 3 ways in which you have used/created “objects “ in Java thus far.

Three ways to create Object in Java are

* Create Class
* We have used keyword “new” to create an object
* Once created class object, it use to access the method inside it using dot operator.
* Using Instances/ instantiations
* Using Constructor to create an object and initialize variables.

1. (5 Points) – Describe a “getter” method. Typically what type of method is used as a “getter” method?

Ans - Getter method is an return method which returns a value. Unlike setter method, we can’t use System.out.println in a getter method. To define getter method, we must first write data type like int, float, double etc and method name with arguments.

1. (5 Points) – Describe a “setter” method. Typically what type of method is used as a “setter” method?

Unlike getter method Setter method is use to set the value. It is not a return method and we can use System.out.println in this method. This is the method where we assign a value to variables. This method also called a void method.

5. (5 Points) Complete the chart, 1 Point per correct entry.

|  |  |
| --- | --- |
| Programming Convention | Programming Syntax |
| Indentation | Semi colons ; |
| Variable names have lower case letters | Open bracket closed bracket {} |
| Capitalize first word of the second letters | When define function, must declare type name (), open and close {} |
| Comments | For the System.out.println(), S must be Capital. |

6. (10 Points) Explain Polymorphism in your own words and give an example used in class.

This is principle which can also be applied to object-oriented programming and languages like the Java language. Subclasses of a class can define their own unique behaviors and yet share some of the same functionality of the parent class.

Polymorphism can be demonstrated with a minor modification to the Bicycle class. For example, a printDescription method could be added to the class that displays all the data currently stored in an instance.

public void printDescription(){

System.out.println("\nBike is " + "in gear " + this.gear

+ " with a cadence of " + this.cadence +

" and travelling at a speed of " + this.speed + ". ");

}

To demonstrate polymorphic features in the Java language, extend the Bicycle class with a MountainBike and a RoadBike class. For MountainBike, add a field for suspension, which is a String value that indicates if the bike has a front shock absorber, Front. Or, the bike has a front and back shock absorber, Dual.

Here is the updated class:

public class MountainBike extends Bicycle {

private String suspension;

public MountainBike(

int startCadence,

int startSpeed,

int startGear,

String suspensionType){

super(startCadence,

startSpeed,

startGear);

this.setSuspension(suspensionType);

}

public String getSuspension(){

return this.suspension;

}

public void setSuspension(String suspensionType) {

this.suspension = suspensionType;

}

public void printDescription() {

super.printDescription();

System.out.println("The " + "MountainBike has a" +

getSuspension() + " suspension.");

}

}

7. (10 Points) Describe Encapsulation in your own words. Describe the primary way encapsulation is used in Java.

Encapsulation is one of the four fundamental OOP concepts. The other three are inheritance, polymorphism, and abstraction.

Encapsulation in Java is a mechanism of wrapping the data (variables) and code acting on the data (methods) together as as single unit. In encapsulation the variables of a class will be hidden from other classes, and can be accessed only through the methods of their current class, therefore it is also known as data hiding.

To achieve encapsulation in Java

- Declare the variables of a class as private.

- Provide public setter and getter methods to modify and view the variables values.

- It is a block of code that restrict access to class methods and instance variables.

The public setXXX() and getXXX() methods are the access points of the instance variables of the EncapTest class. Normally, these methods are referred as getters and setters. Therefore any class that wants to access the variables should access them through these getters and setters.

Benefits of Encapsulation:

* The fields of a class can be made read-only or write-only.
* A class can have total control over what is stored in its fields.

- The users of a class do not know how the class stores its data. A class can change the data type of a field and users of the class do not need to change any of their code.

8. (10 Points) Compare and contrast Strings and Arrays. What do they have in common and how are they different. (In your own words.)

Array and String are true objects in Java. To use an array, we must declare it before we use it.

Array- collection of primitive data type, e.g. int, double, float, char, Boolean

***Declaring an array***

In Java, you must declare an array before you can use it. In declaring the array, you must provide two important pieces of information:

- the name of the array

- the type of data to be stored in it.

We must know the index of array that resembles to a specific data in order to access the data.

Strings- Object – collection of Char – ASCII – represents all characters, symbols, keys on the keyboard as numbers.

String Objects can not be changes once they have been created.

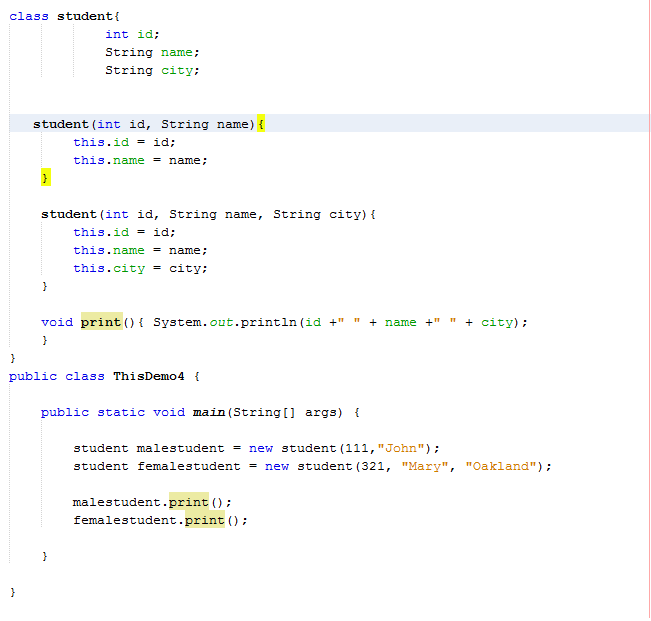
Like array, String also used to holds characters.

Strings, which are widely used in Java programming, are a sequence of characters. In the Java programming language, strings are objects.

Whenever it encounters a string literal in your code, the compiler creates a String object with its value in this case, "Hello world!'.

As with any other object, you can create String objects by using the new keyword and a constructor. The String class has eleven constructors that allow you to provide the initial value of the string using different sources, such as an array of characters.

**DIAGRAM PROGRAMMER CREATED CLASS:** (10 Points – 2 Points Each)



Created two Objects namely malestudetnt and femalestudent.

Used (.) dot operation to access the method called print() in the class called students

Constructor with three arguments, namely int, String, String type

Assigning the value to the variables created in the class.

This is called Arguments, We are making constructor of two arguments namely int and String named as id and name.

Declaring the String name called “name” inside the class.

**SHORT PROGRAMMING TASKS:**

Note: - As we have studied in class, The convention way to declare the class name is to capitalize the first letter of the class, but here in this class, it is not following convention because all letters are small case “studetns”, It should be “Students” instead.

(30 points) TASK 2: Object Orientated Programming

Take your code from Assignmnet#4 Task 1 and Task 2 change the source code to reflect an object orientated model for programming.

Use Constructors, Parameters, and arguments as needed.

Write a program that has 2 methods.

1 method will convert Seconds into minutes.

1 method will convert seconds to hours, minutes, and seconds

Use the modulus arithmetic symbol for part of the calculation/conversion if you like.

**Output=** Show number of seconds, conversion to minutes, and remaining seconds.

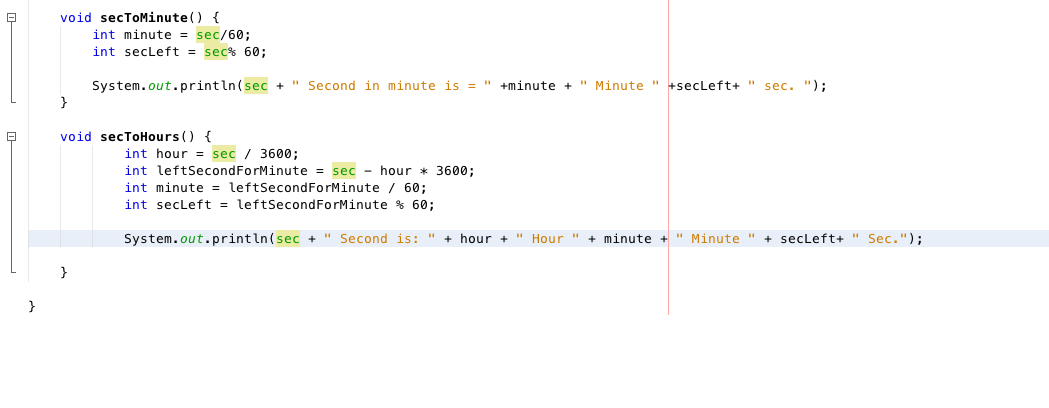
or

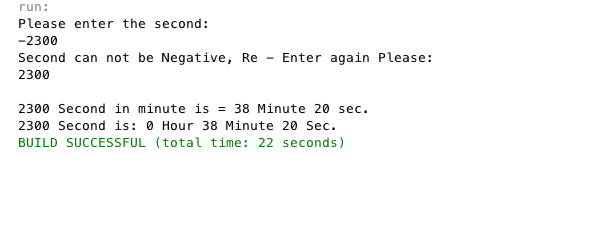
Show number of seconds, converted to hours, converted to minutes, and remaining seconds.

Attach Source Code photo

Attach output of both methods being run.







(20 points) TASK 3:

Use a **recursion** to **print all the numbers from an integer (n) to 100.**

The program should be able to start from any integer 0 – 100, then print the numbers between **(n)** and 100.

Example Output: Source Code ( 2 Snipping Photos: Source Code + expected output)

//Referencing recursion method created in the “program created class.”

public static void main(String[] args) {

numbers sample = new numbers();

sample.numbers(**89**);

}

**Print/Output:**

89

90

91

92

93

94

95

96

97

98

99

100

