LESABA, JOHN KENNETH S.
CSMC211 – N1

1. What are the advantages of using Java's byte-code compilation and Java Virtual Machine compared to straight machine code compilation like C++? How does this relate to the "Write-once, run anywhere" philosophy of Java?

   The main advantage of using Java's bytecode compilation and Java Virtual Machine is that a programmer may compile his/her code on his/her, for example, PC that uses the x86 architecture, and the compiled bytecode (.class extension) can be run on a different processor architecture (such as ARM, etc.) Java compilation does this by compiling the source code and translating it to Java bytecode instead of translating it to a CPU specific machine language. The compiled Java bytecode is then fed into Java's interpreter and that interpreter interprets it to the machine code that the system that it is installed in understands and executes and runs the program in real time. This feature effectively makes Java compiled code compatible to run virtually anywhere.

---

2. Write the complete code for a basic Java program. Explain the function of each part of this program.

```
//public class helloWorld declares the class helloWorld with public visibility
public class helloWorld{

//public static void main is for the declaration of the main method with String array arguments as its argument
  public static void main(String[] args) {

     //This prints the string "Hello, World" in one line.
    System.out.println("Hello, World!");

   // to close the main method
  }
//closing curly brace for the whole class helloWorld
}
```

---

3. Write a Java program that takes in two values: a string MY_NAME and an integer MY_AGE. The program should output “Hello, MY_NAME. Are you MY_AGE years old?” where MY_NAME and MY_AGE are the user-supplied values.

```
import java.util.Scanner;

public class Greetings{
  public static void main(String[] args) {
    Scanner cin = new Scanner(System.in);
    String MY_NAME;
    int MY_AGE;

    System.out.print("Enter name: ");
    MY_NAME = cin.next();
    System.out.print("Enter age: ");
    MY_AGE = cin.nextInt();

    System.out.println("Hello, " + MY_NAME + ". Are you " + MY_AGE + " years old?");
  }
}
```

---

4. Discuss the concept of Object-oriented programming. Differentiate this from the imperative-style used in C/C++. What are some implications of OOP in software development?

The concept of OOP is to dividing parts of the program into different pieces. This effectively makes the program modular. The concept is, in solving a programming problem, with OOP, you can divide and conquer. You can dissect and divide the problem into sub-problems and in solving all the sub-problems, you solve the main problem. OOP makes software development much easier as the sub-problems can be assigned to other programmers in a software development team instead of solving the main problem by only one programmer. This makes software development more efficient.

---

5. Apply the idea of OOP in the game of basketball. Suppose each player is an object of class Player. Define class Player’s state and behavior to the best of your ability. Explain the inclusion of each state element and behaviors.

class Player has the Player constructor that creates players. The Player constructor defines the state and behavior of each player. With the idea of OOP, roles are assigned to different players based on their state and behavior. For example, the created player is very

tall, has a strong built, and behaves rascally in the court, he can be assigned as center. Another example, if the created player is, let us say not the tallest of the bunch, but behaves calmly and can assess situations more than others, he can then be assigned as point guard and team captain. They all have that one goal, to shoot the ball. It is essentially the same as OOP because in OOP, in solving the sub-problems, the main problem is solved.

---

6. Create a Java class called “InstantNoodles” that satisfies the following description:
   a. Objects of this class must be able to store the brand name of the product, its flavor, weight, type (Canton, Ramen, Sotanghon, etc), number of calories.
   b. The brand name of any object from this class must never be blank.
   c. Any InstantNoodle object can be opened and once opened, cannot be closed.
   d. There must be a behavior that checks if the InstantNoodles is closed.
   e. Initially, all InstantNoodles are “Raw”. InstantNoodles can be cooked, represented by the “Cooked” status. If a cooked InstantNoodle is cooked again, then its status must become “Overcooked”.
   f. There must be a behavior that checks the cooking status of any InstantNoodle.

```
public class InstantNoodles{
  private String brand, flavor, type, cookStatus;
  private double weight, cal;
  private boolean open;

  public InstantNoodles(String brandName, String xflavor, String xtype, double xweight, double xcal){
    if(brandName == "")
      brand = "Nissin";
    else
      brand = brandName;
    flavor = xflavor;
    type = xtype;
    weight = xweight;
    cal = xcal;
    open = false;
    cookStatus = "Raw";
  }

  public void openNoodles() {
    if(!open)
      open = true;
  }

  public boolean isOpen() {
```

```
  return open;
 }

 public String cookNoodles() {
  if(cookStatus == "Raw")
   cookStatus = "Cooked";
  else if(cookStatus == "Cooked")
   cookStatus = "Overcooked";
  return cookStatus;
 }

 public String isCooked() {
  return cookStatus;
 }
}
```

---

7. Assume a Java class called Laptop exists with the definition below. Create a complete Java program that instantiates a Laptop object, checks if it is turned on, if its not turned on then turn it on, and finally, login in to the laptop. The output of the program should be “This laptop is functioning normally” upon calling the systemStatus() method.

```
public class testLaptop{
 public static void main(String[] args) {
  Laptop HP = new Laptop();

  if(!HP.isOn())
   HP.turnOn();

  HP.login("default","1234");

  HP.systemStatus();
 }
}
```