COSC 1436: Programming Fundamentals I
Lab 10: Methods and Objects

Please submit one source code file (.java file) for each exercise. (Ex. Lab10Ex1.java, Lab10Ex2,java, Lab10Ex3.java ...etc).

Remember to follow the commenting guidelines.

Program 1: Creating objects (40 points):

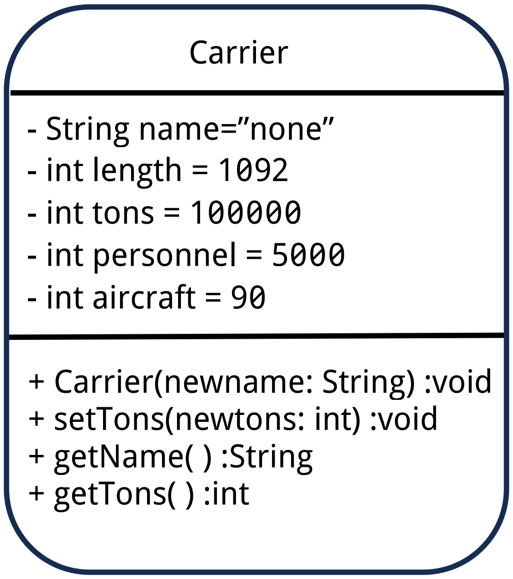
Imagine you are working on a fleet simulation for the US Navy. Each ship will be an instance of its ship type.

Define a class with the following fields, methods, and constructor. Watch out to make the specified fields and methods `private` instead of `public`.

Using that class, create two instances of carriers:

1. Instance: CVN68
   - Name=”Nimitz”

2. Instance: CVN77
   - Name=”George H.W. Bush”
   - Tons=102000



Carrier

- String name=”none”
- int length = 1092
- int tons = 100000
- int personnel = 5000
- int aircraft = 90

+ Carrier(newname: String) :void
+ setTons(newtons: int) :void
+ getName( ) :String
+ getTons( ) :int

Submit both the .java file with the Carrier class, and the .java file with your main program.

```
public class Prog1{
  public static void main(String[] args){
    Carrier CVN68=new Carrier("Nimitz");
    Carrier CVN77=new Carrier("George H.W. Bush");
    CVN77.setTons(102000);
    System.out.println(CVN68.getName()+"
"+CVN68.getTons());
    System.out.println(CVN77.getName()+"
"+CVN77.getTons());
  }
}
```

```
public class Carrier{
   private String name="none";
   private int length=1092;
   private int tons=100000;
   private int personnel=5000;
   private int aircraft=90;

   public Carrier(String newname){
     name=newname;
   }

   public void setTons(int newtons){
     tons=newtons;
   }

   public String getName(){
     return name;
    }

   public int getTons(){
     return tons;
   }
}
```

Program 2: Dice game (40 points):

Write a game that you play against the computer that involves each of you rolling three 6-sized dice.

1. For each round:
   - Roll three 6-sided dice and add up the sum.
   - Have the computer do the same
   - Whoever has the highest sum wins the round
   - If there is a tie, there is no winner.
2. After each round, print out how many rounds you have won and how many the computer has won.
3. Program should ask if you want to play another round. Keep playing as long as you answer 'Y'. Print "Goodbye" if you answer anything else.

For this program, create a method to roll the three 6-sided dice (use `Random`) and return the sum of the rolls. Use this method when you calculate the computer's roll and also when you calculate your roll.

Example:

```
Computer rolls 12
You roll 9
Computer wins!
Score: You=0  Computer=1
Would you like to play again? Y/N Y

Computer rolls 11
You roll 15
You win!
Score: You=1  Computer=1
Would you like to play again? Y/N Y

Computer rolls a 12
You roll a 12
Tie! No winner this time
Score: You=1  Computer=1
Would you like to play again? Y/N r

Goodbye
```

```java
import java.util.Random;
import java.util.Scanner;

public class Program2{
  public static void main(String[] args){
      Scanner kb=new Scanner(System.in);
      int computer=0;
      int user=0;
      int computerscore=0;
      int userscore=0;
      char again='A';

      do{
        computer=roll();
        user=roll();
        System.out.println("Computer rolls "+computer);
        System.out.println("You roll "+user);
        if (computer>user){
            System.out.println("Computer wins!");
            computerscore++;
        }
        else if (computer<user){
            System.out.println("You win!");
            userscore++;
        }
        else {
          System.out.println("Tie! No winner this time");
        }
        System.out.println("Score: You="+userscore+"
Computer="+computerscore);
        System.out.println("Would you like to play again? Y/N");
        again=kb.nextLine().charAt(0);
      }while (again=='Y');
      System.out.println("Goodbye");
  }

  public static int roll(){
    Random rand = new Random();
    int die1=rand.nextInt(6)+1;
    int die2=rand.nextInt(6)+1;
    int die3=rand.nextInt(6)+1;

    return (die1+die2+die3);
  }
}
```

Multiple Choice Questions (20 points)

Please include your answers to the multiple choice questions in comments at the end of your 3rd program from this lab.

1. Which of the following describes this passing scheme?

   When a method is called, a new memory location is set aside and assigned to each of the inputs sent to the method. Values of the inputs are copied into the new locations and the method only makes changes to these new locations. When the method finishes executing, values in these new memory locations are erased. Main memory locations are not changed, except for the `return` value.

   a) Pass by value
   b) Pass by reference
   c) Pass by name

2. Which of the following describes this passing scheme?

   When a method is called, a new memory location are not assigned for the inputs. Instead, the method just uses the same memory locations as the main method. Any changes that the method makes to the memory location affects the main method too.

   a) Pass by value
   b) Pass by reference
   c) Pass by name

3. How does Java pass an array to a method?

   a) Pass by value
   b) Pass by reference
   c) Pass by name

4. How does Java pass a regular variable to a method?

   a) Pass by value
   b) Pass by reference
   c) Pass by name

5. What are the 4 main principles of Object-Oriented Programming?

    a) Abstraction, Encapsulation, Inheritance, Protection
    b) Classes, Objects, Private, Fields
    c) Objects, Orientation, Programming
    d) Abstraction, Encapsulation, Inheritance, Polymorphism

6. When you make a field in a class `private`, it can accessed by

    a) Any method
    b) Only the main method
    c) Only methods defined in the class
    d) Only other methods that are `private`

7. Can one class have multiple constructors?

    a) No
    b) Just the default and one other
    c) Any number, no restrictions
    d) Any number, but each constructor must have a different list of input parameters

8. What is the difference between a class and an object?

    a) A class is a “template” that describes the variables and methods, while the object is the “template” actually created in memory
    b) An object is a “template” that describes the variables and methods, while the class is the “template” actually created in memory
    c) An object is a subset of class
    d) No difference. Just two different words for the same thing.