# Practice Exam 2 - COP 3502 Spring 2022

## **Module 4B:** Arrays

Q1. What is the output of the following program?

```
public static void main(String[] args) {
    int[] a = {11, 22, 10, 8};
    int[] b = a;
    a = new int[] {3, 13, 12, 15};
    a[3] = 2;
    for (int i : b)
        System.out.print(i + " ");
}
```

a. 11 22 10 8<br>
b. 3 13 12 15<br>
c. 11 22 10 2<br>
d. 3 13 12 2<br>
e. Error

Q2. Which of the following would return null?

```
public class Matrix{
    public static void main(String[] args) {
          int[][] matrix = new int[5][3];
          matrix[0] = new int[] {1, 3, 4};
          matrix[1] = new int[5];
          matrix[2] = new int[] {12, 23, 11};
    }
}
```

a. matrix[3]<br>
b. matrix[1][2]<br>
c. matrix[4]<br>
d. matrix[3] and matrix[4]<br>
e. None of the above

Q3. Which of the following would return null?

```
public class Matrix{
    public static void main(String[] args) {
          int[][] matrix = new int[5][];
          matrix[0] = new int[4];
          matrix[1] = new int[6];
          matrix[2] = new int[2];
          //insert array access
    }
}
```

A. matrix[0]

B. matrix[1][1]
C. matrix[2][1]
D. matrix[3]
E. None

Q4. What is the output of the following program?

```
public static void main(String[] args) {
       int[][] arr = new int[4][];
       arr[0] = new int[2];
       arr[1] = arr[0];

       arr[0][0] = 3;
       arr[1][1] = 4;
       for (int i = 0; i < arr[0].length; i++) {
         System.out.print(arr[0][i] + " ");
       }
}
```

A. 3 4
B. 0 0
C. 3 0
D. null null
E. None of the above

## **Module 5**: Version Control and Exception

Q1. Which of the following commands will stage your entire directory and every non-empty directory inside your current directory?

a. git status all
b. git add .
c. git add all
d. git commit all

Q2. Which of the following statements is not correct?

a. The commit command is used to save your changes to the local repository.
b. Commits are local until they are pushed.
c. A clone operation makes a copy of a repository locally.
d. A push operation pulls down any changes made to the remote repository.

Q3. What is the output of the following program?

```
public static void main(String[] args) {
  try {
      int num = 4;
      num = Integer.parseInt("Interesting");
      System.out.println(num);
```

```
        }
        catch(Exception e) {
            System.out.println("Something Wrong");
        }
      }
```

a. Interesting

b. Something Wrong

c. 3

d. Error

## **Module 6:** Class

Q1. We have a class Monkey that is defined as follows:

```
public class Monkey{
      private String name;
      private int height;
      public Monkey () {}
}
```

We define mutator methods to set the values of name and height variables of a Monkey object.

1) The return type of the mutator or setter function for the height instance variable is?

a. int
b. double
c. String
d. void

2) The return type of the accessor or getter function for the height instance variable is?

a. int
b. double
c. String
d. void

Q2. What is the output of the following code?

```
public class Student {
      public int grade;

      public Student(int g) {
            grade = g;
      }
```

```
    public static void increaseGrades(int[] grades, Student  student,
    int grade) {
        grades[0] += 25;
        student.grade = 100;
        grade = 101;
    }

    public static void main(String[] args) {
        int[] grades = {85, 80, 90};
        Student bob = new Student(grades[1]);
        int gradeX  = grades[2];
        increaseGrades(grades, bob, gradeX);
        System.out.println(grades[0] + bob.grade + gradeX);
    }
}
```

300

Q3. Given the following class definition,

```
public class Hero {
    String name;
    int power;
    static int heroCount = 0;

    public Hero(String name, int power) {
        this.name = name;
        this.power = power;
        heroCount++;
    }

    public static void death(int deathCount) {
        heroCount -= deathCount;
    }
}
```

What is the output of the following code snippet?
*Assume that the main method is implemented and ran properly*

```
public static void main(String args[]) {
    Hero jim = new Hero(“jim”, 12);
    Hero pam = new Hero(“pam”, 9);
    Hero andy = new Hero(“andy”, 4);
    jim.heroCount--;

    Hero erin = new Hero(“erin”, 10);
    Hero michael = new Hero(“michael”, 7);

    erin.death(2);

    System.out.println(Hero.heroCount);
```

2

```
}
```

Q4. What is the output of the following program?

```
public class Apple {
    private double weight;
    public Apple(double weight) {
        this.weight = weight;
    }
    public double getWeight() {
        return weight;
    }
    public void setWeight(int _weight) {
       weight = _weight;
    }
}
public class Main {
    public static void main(String[] args) {
        Apple fuji = new Apple();
        System.out.println(fuji.getWeight());
        fuji.setWeight(3);
    }
}
```

a. 0
b. 3
c. null
d. Error

Q5. What is the output of the following program? (If the program is not able to be executed, please indicate "Error".)

```
public class Apple {
    private double weight;
    public Apple(double weight) {
        this.weight = weight;
    }
    public double getWeight() {
        return weight;
    }
    public void setWeight(int _weight) {
       weight = _weight;
    }
}
public class Main {
    public static void main(String[] args) {
        Apple fuji = new Apple(14);
        System.out.println(fuji.getWeight());
        fuji.setWeight(3);
        System.out.println(fuji.getWeight());
```

14
3

```
        }
    }
```

## **Module 7:** Inheritance

Q1. Given the following class names and assuming they all have default constructors:

```
    class A {...}
    class B extends A {...}
    class C extends A {...}
    class D extends C {...}
    class E extends B {...}
```

Which of the following code are allowed?

a. A myClass = new A(); ✓
b. E myClass = new E(); ✓
   B nextClass = myClass; ✓
c. C myClass = new A();  //no ✗
d. Object myObject = new E(); ✓
e. B otherClass = new C(); //nope ✗
f. Object myClass = new A(); ✓
   A secondClass = (A)myClass; ✓
g. A thirdClass = new C(); ✓
   C nextClass = (C)thirdClass; ✓

Q2. What is the output of the above program?

```
    public class Fish {
        int weight;
        public void swim() {
            System.out.println("Splash");
        }
    }
    public class Salmon extends Fish {
        String home;
        public void swim() {
            System.out.println("Splish splash");
        }

        public void swim(int speed) {
            System.out.println("swimming at " + speed + " mph");
        }
    }
    public class Main {
        public static void main(String[] args) {
            Fish fish = new Fish();
            Salmon salmon = new Salmon();
            Fish mike = new Salmon();
            salmon.swim();
            mike.swim();
```

splish splash
splish splash

```
            mike.swim(5);   Error
        }
    }
```

Note: Fish class is defined in the Fish.java file
Salmon class is defined in the Salmon.java file
Main class is define in the Main.java file

Q3. Predict the output of the numbered lines in the main method. //1, //2, etc...
Write the proper output for each line.
If a line would cause an error, assume that the program would continue and write"Error".
Example answer:
1. Error
2. Output...

```
    public class Student extends Person {
        public void printType() {
            System.out.println("Student");
        }
        public void printMajor() {
            System.out.println("Computer Science");
        }
    }

    public class Person {
        public void printType() {
            System.out.println("Normal Person");
        }

        public static void main(String[] args) {
            Person a = new Person();
            Person b = new Student();
            Object c = new Student();

            a.printType();                 //1 NP
            ((Student)a).printMajor();     //2 E
            b.printType();                 //3 S
            b.printMajor();                //4 E
            ((Student)b).printMajor();     //5 CS
            c.printType();                 //6 E
            c.printMajor();                //7 E
            ((Person)c).printType();       //8 S
            ((Student)c).printType();      //9 S
            ((Person)c).printMajor();      //10 E
            ((Student)c).printMajor();     //11 CS
            Student d = new Person();      //12 E
        }
        }
```

Q4. What is the output of the following program?
Note: Classes are in separate files.

```
        public class Grandparent {
            public Grandparent() {
                System.out.println("Grandparent constructor");
            }
        }
```

Point

```
        public class Parent extends Grandparent {
            public Parent() {
                System.out.println("Parent constructor");
            }
            public Parent(int age) {
                System.out.println("Parent constructor to set up age");
            }

        }
```

Point

```
        public class Child extends Parent {
            public Child() {
                System.out.println("Child constructor");
            }
            public Child(int age) {
                System.out.println("Child constructor to set up age");
            }
        }
```

Point

```
        public class ConstructorDemo {
            public static void main(String[] args) {
                Child child = new Child(18);
            }
        }
```

Q5. What is the output of the following program?

```
        public class Animal{
             public void print(){
                    System.out.println("Animal printed");
             }
        }
        public class Dog extends Animal{
             public void print() {
                   System.out.println("Dog printed");
             }
             public void print(String name){
                   System.out.println("Hi, my name is " + name + " ...WOOF!");
             }
        }
        public class myAnimals{
             public static void main(String[] args){
                   Animal doggo = new Dog();
                   doggo.print("Carl");
              }
        }
```

a. Animal printed
b. Dog printed
c. Hi, my name is Carl...WOOF!
d. Error

Q6. Assume that the following two classes are valid

```
     public class Apple {// class definition}
     public class Apple extends Fruits {// class definition}
```

The following statement is declared in a main method of another class.

```
     Apple fuji = new Apple();
```

Which of the followings would be the correct way to check the class membership of Apple fuji?

a. fuji is an Apple x
b. fuji instanceof Apple
c. fuji is an Object x instance
d. fuji instanceof Object
e. Apple instance of Fruits x note: It's self
f. Apple instance of Object
g. fuji extends Fruits x Apple

## Coding

Q1. Write a function rotateArray(int[] arr, int d) that rotates the array arr by d elements to the left. The function header for the method is as follows:

```
     public static void rotateArray(int[] arr, int d) {

     }
```

Example:

Before calling rotateArray function, arr = {1, 2, 3, 4, 5}, d = 2
After calling rotateArray function, arr = {3, 4, 5, 1, 2}

**Note:** Only write the method and any helper methods, no need to create a class or create a main method.

Q2. A common statistic of interest is the longest sequence of some pattern in a list of items. For example, in football, one may be interested in the longest sequence of consecutive complete passes. Given an array of strings in which each item is either the letter "I" (for an incomplete pass) or a number (for a completed pass), output the length of the longest sequence of complete passes. Write a method that takes in an array of strings and return the longest sequence.

The function header for the method is as follows:

```
public static int longestSequence(String[] listItems) {

}
```

Example:

Input: listItems = {“4”, “15”, “9”, “I”, “30”, “2”, “I”, “20”}

Return value is 3 (because the longest sequence of complete passes is “4”, “15”, “9”).

Q3. Write a function findDuplicates(int[] arr) that finds whether there are duplicates in the arr. If there is duplicate, return true. Otherwise, return false. The function header for the method is as follows:

```
public static boolean findDuplicates(int[] arr) {

}
```

Example:

Before calling rotateArray function, arr = {1, 2, 3, 4, 5, 3, 1}
After calling rotateArray function, return true

**Note:** Only write the method and any helper methods, no need to create a class or create a main method.

Q4. Given two integers that represent the miles to drive forward and the miles to drive in reverse as user inputs, create a SimpleCar object that performs the following operations:

- Drives input number of miles forward
- Drives input number of miles in reverse
- Honks the horn
- Reports car status

The only program you need to implement is SimpleCar class. The Main class that uses SimpleCar class is provided below:

```
import java.util.Scanner;

public class LabProgram {
   public static void main(String[] args) {
      Scanner scnr = new Scanner(System.in);
      int forward;
      int reverse;
      SimpleCar car = new SimpleCar();

      forward = scnr.nextInt();
      reverse = scnr.nextInt();
      car.drive(forward);
      car.reverse(reverse);
      car.honkHorn();
      car.report();
   }
}
```

Example:

If the input is:

```
100 4
```

the output is:

```
beep beep

Car has driven: 96 miles
```

Q5. Given main(), define the Team class (in file Team.java). For class method getWinPercentage(), the formula is teamWins / (teamWins + teamLosses)

Note: Use casting to prevent integer division.

Ex: If the input is:

Ravens

13

3

where Ravens is the team's name, 13 is number of team wins, and 3 is the number of team losses,

the output is:

Congratulations, Team Ravens has a winning average!

If the input is Angels 80 82, the output is:

Team Angels has a losing average.

The only program you need to implement is Team class. The Main class WinningTeam.java that uses Team class is provided below:

```java
import java.util.Scanner;

public class WinningTeam {
   public static void main(String[] args) {
      Scanner scnr = new Scanner(System.in);

      Team team = new Team();

      String name = scnr.next();
      int wins = scnr.nextInt();
      int losses = scnr.nextInt();

      team.setTeamName(name);
      team.setTeamWins(wins);
      team.setTeamLosses(losses);

      if (team.getWinPercentage() >= 0.5) {
         System.out.println("Congratulations, Team " +
team.getTeamName() + " has a winning average!");
      }
      else {
         System.out.println("Team " + team.getTeamName() +
                            " has a losing average.");
      }
   }
}
```