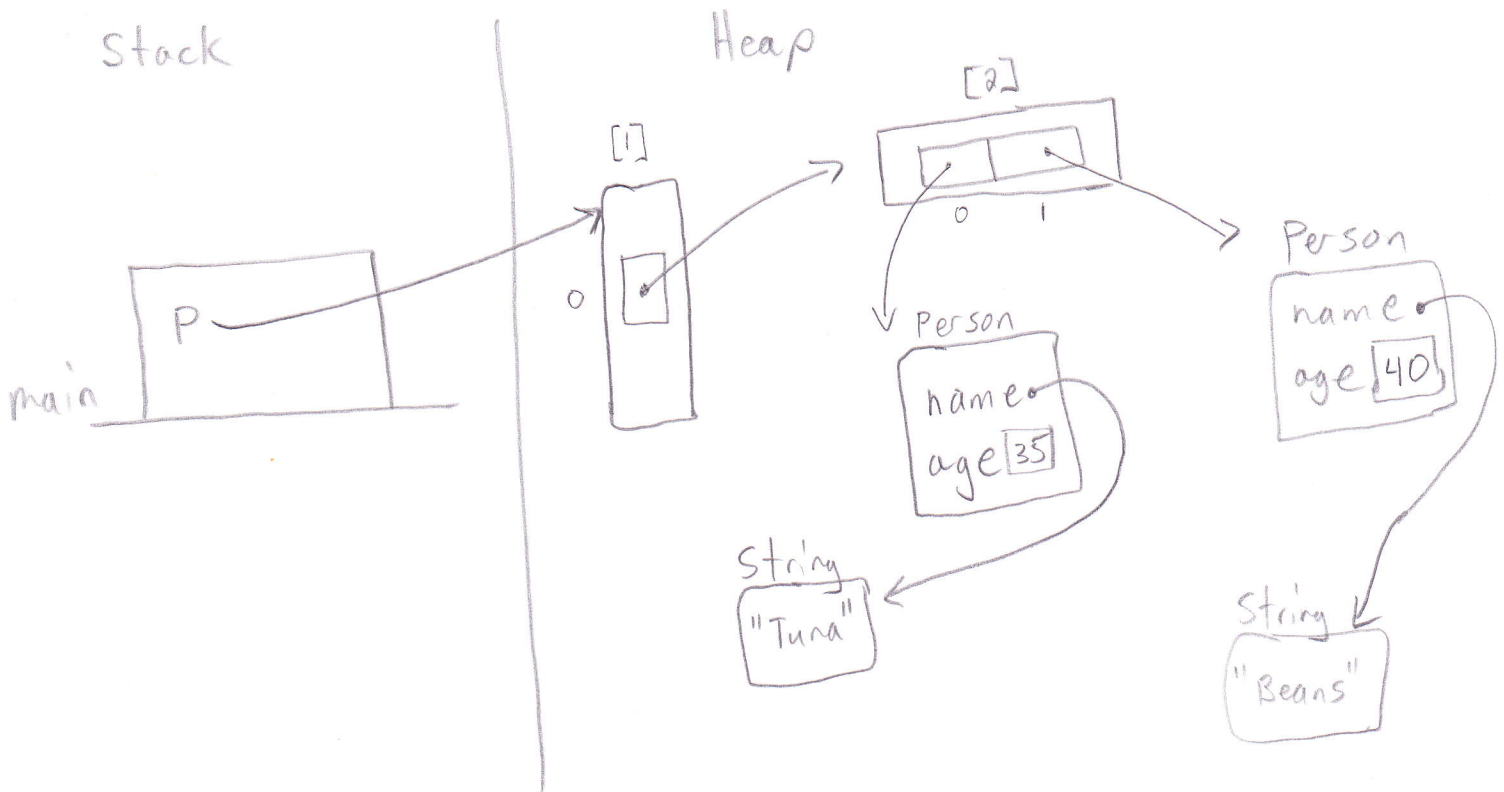


Part 2: Short Answer Questions 30 points [points shown for each question, spend time wisely]

1. Using the code provided draw a memory map to represent Stack and Heap just at the end of method main. [5 points]

```
public class Person {  
    private String name;  
    private int age;  
    public Person(String name, int age){ this.name = name; this.age = age;}  
    public Person(){ this("Person 1", 25); }  
    public String getName(){ return this.name; }  
    public void setName(String name){ this.name = name; }  
    public int getAge(){ return this.age; }  
    public void setAge(int age){ this.age = age; }  
}
```

```
public class PersonDemo{  
    public static void main(String[] args){  
        Person[][] p = new Person[1][2];  
        p[0][0] = new Person("Tuna", 35);  
        p[0][1] = new Person("Beans", 40);  
        // draw memory map from here  
    }  
}
```

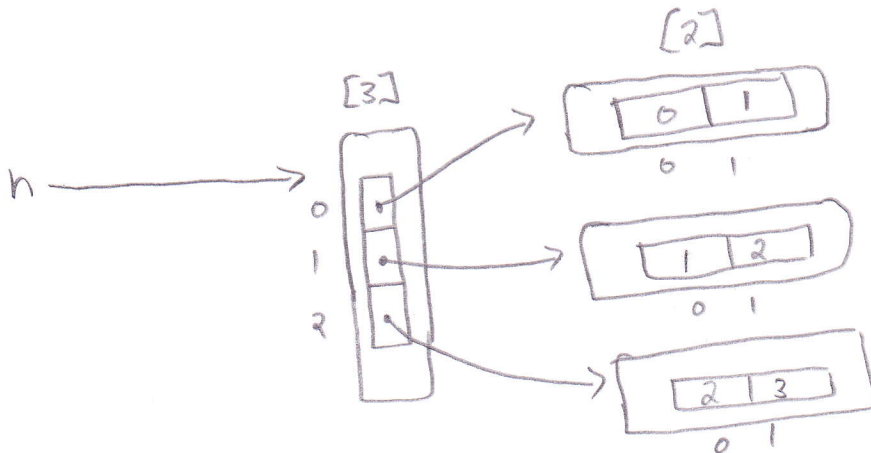


2. Write a loop so that the array shown will have its elements set to the value of adding the indexes, e.g. element at index [0][1] will be set with a value of 1, element at index of [2][1] will be set to 3 (etc.) [4 points]

```
int[][] n = new int[3][2];
```

```
for (int i = 0; i < n.length; i++) {  
    for (int k = 0; k < n[i].length; k++) {  
        n[i][k] = i + k;  
    }  
}
```

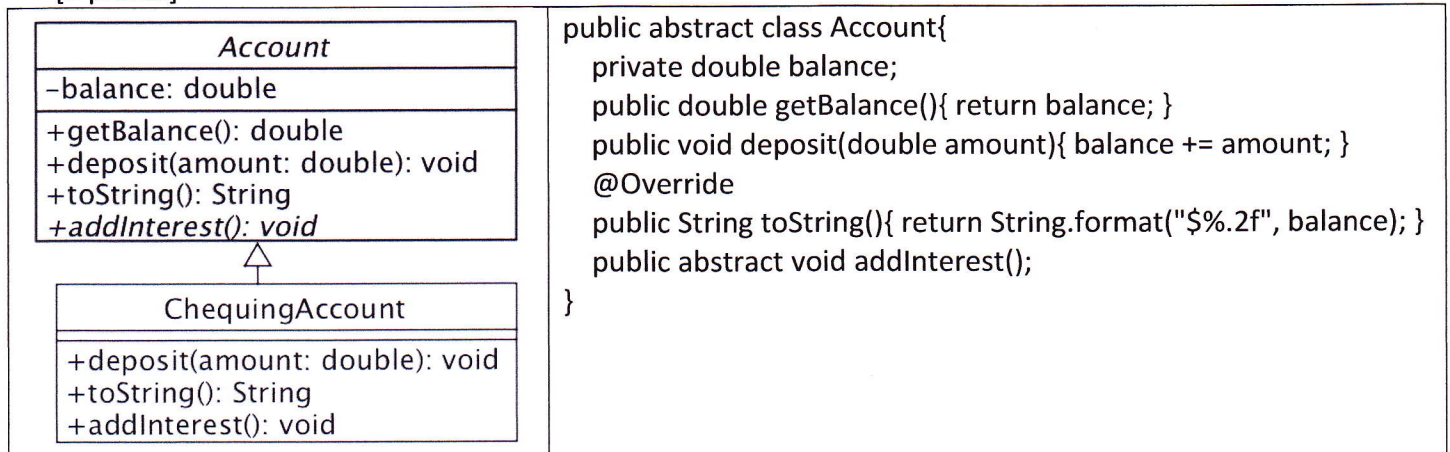
3. Draw a memory map of the array from question 2 after it has been initialized with the requested numbers. [6 points]



4. Examine the figure below as well as the code for class Account then write the code for class ChequingAccount:

- ChequingAccount deposit should deduct 5 cents from the deposit for a service fee
E.g. deposit 50 cents, balance only increases 45 cents
- Expected output from ChequingAccount toString is: Chequing Account Balance: \$0.00
(Assuming the account was created but has a balance of zero)
- addInterest() should add 5% of the current balance to the balance as a deposit.

[5 points]



Tip: Don't forget extends, super., StringBuilder and @Override

```
public class ChequingAccount extends Account {
    @Override
    public void deposit(double amount) {
        super.deposit(amount - 5);
    }
}
```

```
@Override
public String toString() {
    StringBuilder b = new StringBuilder();
    b.append("Chequing Account Balance:");
    return b.append(super.toString());
}
```

```
@Override
public void addInterest() {
    // super.deposit or deposit acceptable here
    super.deposit(getBalance() * 0.05);
}
```

5. Using the words presented as a guide write a JUnit test method to test that methods `getWeight()` and `setWeight(double)` of class `Apple` work correctly. (Each word may be used more than once.)
[5 points]

Apple
-weight: double
+getWeight(): double +setWeight(weight: double): void

Words

@Test	new	apple	expected	actual	double	testWeightGetSet
Apple	Apple()	getWeight	setWeight	assertEquals	delta	0.01
String	message	"Apple weight get set not working"				

```

@Test
public void testWeightGetSet() {
    Apple apple = new Apple();
    double expected = 4.0;
    double delta = 0.01;
    String message = "apple weight get set not working";
    apple.setWeight(expected);
    double actual = apple.getWeight();
    assertEquals(message, expected, actual, delta);
    // see multiple choice #23.
}

```

6. Using the words provided as a guide and the starter code re-write the main method presented so that if the user enters the phrase "tuna fish" the program will report the exception's message and then exit rather than crashing. Note: Need to use exception handling.
(Most of the words below only need to be used once)

[5 points]

```
import java.util.Scanner; // no need to rewrite the import
```

```
public class AgeProgram{
    public static void main(String[] args){
        Scanner s = new Scanner(System.in);
        System.out.println("What is the age?");
        int n = s.nextInt();
        System.out.println("Age entered as: " + n);
    }
}
```

Words

try	Exception	getMessage	catch	e
----------------	----------------------	------------	------------------	--------------

```
public static void main(String[] args) {
    try {
        Scanner s = new Scanner(System.in);
        System.out.println("What is the age?");
        int n = s.nextInt();
        System.out.println("age entered as: " + n);
    } //end try.
    catch (Exception e) {
        System.out.println(e.getMessage());
    } //end try-catch.
} //end main.
```