Class 4

1.

- Build a class with the following line inside main:
  Int a = 1/0;

- Build a corresponding try-catch block to avoid exception.
- Is the following code legal? try {

  ...

  } finally {

  ... }

- What exception types can be caught by the following handler?

  catch (Exception e) {

  ... }

- What is wrong with using the above type of exception handler?
- What exceptions can be caught by the following handler?

  ...

  } catch (Exception e) {

  ...

  } catch (ArithmeticException a) {

  }

2.

- Create an interface with 1 method.
- Implement the interface and do something with the method.

3.

- Create an abstract class with 1 method.
- Extend the abstract class and do something with the implemented method.

4.

- What will be the result of the following code:

```
abstract class AbstractClass {
void concreteMethod() { // concrete methods are still allowed in abstract classes
System.out.println("This is a concrete method.");
   }
}

public class Main {
public static void main() {
      AbstractClass class = new AbstractClass();
   }
}
```

5.

- Create an abstract class and declare it final class
- Add an abstract method to class
- What will be the result and why?

6.

Write a program with two classes:

- Create a class named Ball
- declare a static final boolean called “isRound” with true value.
- Print the variable value from Main.class
- Try to change the value from Main.class

7.

What is the issue with the following code:

```
public static void test(){
static int x = 3;
}
```

8.

What is wrong with the below program:

```
public class Main {
    public static void main(String[] args) {
        String s = null;
        System.out.println(s.contains("a"));
    }
}
```

9.

When should we call garbage collection?

10.

```
public class Cow extends Animal implements Inter{
}
```

Cow is an instance of:

A. Cow
B. Animal
C. Inter
D. All the above.

11.

- Does it make sense to use a try-finally block (without catch)?
- When?

12.

What will be the below program output?

```
public class Main {
    public static void main(String[] args) {
        int x = 5;
        if (x == 5){
            throw new RuntimeException();
        }
        System.out.println(555);
    }
}
```

Challenges:

13.

- What is the reserved keyword **const** purpose in Java?
- Can it be used?

14.

- Create an exception class
- Throw it (anyhere)