

### Objectives:

- Write Programs that make uses abstract classes, interfaces and Exception Handling

### Exercise 1

### Compulsory

We will modify an existing program to convert it into an abstract class. The code is at

<https://goo.gl/gbMxU6>

// Lab Sheet on Abstract Classes

// Starter Code

```
class Animal {
    private String name;
    public Animal(String name) {
        this.name = name;
    }
    public String speak() {
        return "";
    }
    public void display() {
        System.out.println("My name is " + this.name + ". " + this.speak() + ".");
    }
}

class Dog extends Animal {
    public Dog(String name) {
        super(name);
    }
    public String speak() {
        return "Bow Wow";
    }
}

class Cat extends Animal {
    public Cat(String name) {
        super(name);
    }
    public String speak() {
```

---

```
        return "Meow Meow";
    }
}

class ToyCat extends Cat {
    String manufacturer;
    public ToyCat(String name, String manufacturer) {
        super(name);
        this.manufacturer = manufacturer;
    }
    public void display() {
        super.display();
        System.out.println("I am from " + manufacturer + ".");
    }
}

class Main {
    public static void main(String[] args) {
        Animal animal1 = new Animal("test");
        animal1.display();

        Cat mycat = new Cat("Micky");
        mycat.display();

        Dog mydog = new Dog("Rover");
        mydog.display();

        ToyCat mytoy = new ToyCat("kittie", "Toysrus");
        mytoy.display();
    }
}
```

- a) Make the Animal class an abstract class and make the speak method an abstract method. Modify the main function.

What needs to be removed and why?

- b) Add default constructors to each of the classes. Display a message giving details of the name of the constructor.

e.g.

```
public Animal() {  
    System.out.println("Animal constructor called");  
}
```

Create an object of the ToyCat class and call the default constructor. What will be the output? Can you explain how why the output appears as displayed?

**Exercise 2****Compulsory**

We will write two classes that implement a given interface. The code is at <https://goo.gl/7bXxem>

// Interface Lab Sheet

```
interface ICompute {  
    void calculate();  
    void display();  
}
```

```
class Person {  
    private String name;  
    private double basicSal;  
    private double otRate;  
    private double otHrs;  
    private double netSal;  
}
```

```
class Box {  
    private int length, width, height;  
    private int volume;  
}
```

```
class Main {  
    public static void main(String[] args) {  
        ICompute ob1 = new Person("Danushka", 40000, 1000, 5);  
        obj1.calculate();  
        obj1.display();  
  
        ICompute ob2 = new Box(10, 20, 30);  
        obj2.calculate();  
        obj2.display();  
    }  
}
```

- a) Implement the Person making use of the ICompute interface
  - i) Write a constructor
  - ii) implements the methods in ICompute. In the compute() method calculate netSal
- b) Implement the Box Class making use of the ICompute interface
  - i) Write a constructor.
  - ii) implements the methods in ICompute. In compute() calculate the volume
- c) Create variables of Box and Person Type and create objects calculate and display values.
- d) Describe any advantage you see in using interface type variables seen in the original main function code variables obj1, obj2 as opposed to using object type variables

**Exercise 3**

We will write a program that will allow us to enter a given number of marks and calculate the average. We will use a try catch block in the code. The code is at <https://goo.gl/i6F3Pq>

Complete the comments given in the code. See the following links for specific exceptions that you can use.

<https://docs.oracle.com/javase/8/docs/api/java/util/InputMismatchException.html>

<https://docs.oracle.com/javase/8/docs/api/java/lang/ArithmeticException.html>

<https://docs.oracle.com/javase/8/docs/api/java/lang/ArrayIndexOutOfBoundsException.html>

// Lab Sheet on Try Catch Blocks

```
class Main {  
    public static void main(String[] args) {  
  
        int maxSubjects;  
        int [] marks = new int[5];  
        int total;  
        double avg;  
  
        try {  
            // 1. Input a value for maxSubjects  
            //   from keyboard  
            // 2. Using a for loop  
            //   input marks  
            // 3. Calculate the avg marks  
            // 4. Use a try catch block to  
            //   prevent the following  
            //   run time errors  
            //   (a) Input valid integers to the  
            //       inputs  
            //   (b) ArithmeticException division  
            //       by zero  
            //   (c) ArrayIndexOutOfBounds  
            //       Exception
```

---

```
} catch (Exception e) {  
  
}  
  
finally {  
    System.out.println("This code will be gurrentied to run");  
}  
  
System.out.println("The end");  
  
}  
}
```