13/06/2020 Java

- Generics and Wildcards:

```
                ArrayList<Machine> List1 = new Array list<Machine>();
               List1. add(new Machine());
               list1. add(new Machine());
          ArrayList <Camera> List 2=new Arraylist <camera>();
            List2. add (new Camera());
           List2. add (new Camera());
            showList(list2);
            Showlist 2(list1);
            Showlist 3(List1);
           }
```

- Anonymous Classes:

```
        Machine machine1 = new Machine(){
        @Override public void start(){
            System.out. println("Camera Snapping...");
          }
        };
          machine1.start();
          plant plant1= new Plant(){
            @Override
              public void grow(){
                 System .out. println("plant growing);
              }
           };
            plant1.grow();
        }
       }.
```

- Reading files using scanner:

```
                    File textfile = new filename
                      Scanner in = new Scanner (textfile);
```

Scanned with CamScanner

- **Handling ~~Expec~~ Exceptions:**

```
public static void main (string[]args){
 try{
     openFile();
 }catch (File Not found Exception e){
   // PS. This message is too vague :)
     System.out.println ("Could not open file");
}
}
public static void openfile() throws File not Found
                              Exception{
```

- **Multiple Exceptions:**

```
  try{
      test.run();
  }catch(Exception e){
    //TODO Auto-generated catch block
      e.printStackTrace();
  }
  try{
     test.input();
  }catch (File Not Found Expection e){
  } catch (IO Exception e) {
      e.printstackTrace();
```

- **Runtime vs. Checked Exceptions:**

```
Public class App{
   Public static void main (string[] args){
   string[] texts = {"one", "two", "three"};
  try {
     System.out. println(texts(3));
  }catch( ArrayIndexOut of BoundException e)}{
  }
 }
}
```

• Abstract Classes

```
public class Camera extends Machine{
@Override
 public void start(){
   System.out. printlm("Starting Camera");
}
 @Override
  public void dostuff
   //TODO Auto-generated method stub
public abstract void ~~str~~ strat();
public abstract void dostuff();
public abstract void shutdown();
 public void run()
    start ();
    dostuff();
    Shutdown();
 }
}
```

• Reading Files with File Reader:

```
BufferedReader br = null;
try{
   file reader fr = new file Reader (file);
   br = new BufferedReader (fr);
string line;
while(line = br.readline()) != null ){
  System.out.println (line);
}
```

• Try with resources.

• Creating and Writing Text Files:

```
    File file = new file ("test txt");
try (BufferedWritter br = new bufferedWritter (new
        fileWriter(file)))
   br.write (" This is line one");
   br.newline();
   br.write("This is line two");
   br. new line();
  }.
```