

# Notes

Introduction to Computer Science (CS50) on EdX

Sparsh Jain

December 14, 2020

# Contents

|          |  |           |
|----------|--|-----------|
| <b>I</b> | <b>General</b>                         | <b>6</b>  |
| <b>1</b> | <b>Computational Thinking, Scratch</b> | <b>6</b>  |
| 1.1      | Binary Number System . . . . .         | 6         |
| 1.2      | Algorithms . . . . .                   | 6         |
| 1.3      | Time Complexity . . . . .              | 6         |
| 1.4      | Pseudocode . . . . .                   | 6         |
| 1.5      | Scratch . . . . .                      | 6         |
| <b>2</b> | <b>C</b>                               | <b>7</b>  |
| 2.1      | Hello World . . . . .                  | 7         |
| 2.2      | Input . . . . .                        | 7         |
| 2.3      | Initialization . . . . .               | 9         |
| 2.4      | Increment . . . . .                    | 9         |
| 2.5      | Conditionals . . . . .                 | 9         |
| 2.6      | Loops . . . . .                        | 9         |
| 2.6.1    | While Loop . . . . .                   | 9         |
| 2.6.2    | For Loop . . . . .                     | 10        |
| 2.7      | Additional Info . . . . .              | 10        |
| 2.7.1    | Datatypes . . . . .                    | 10        |
| 2.7.2    | Functions . . . . .                    | 10        |
| 2.7.3    | Placeholders . . . . .                 | 11        |
| 2.7.4    | Arithmetic Operations . . . . .        | 11        |
| 2.8      | Examples . . . . .                     | 11        |
| 2.8.1    | Arithmetic . . . . .                   | 11        |
| 2.8.2    | Conditional . . . . .                  | 14        |
| 2.8.3    | Logical . . . . .                      | 15        |
| 2.8.4    | Loop . . . . .                         | 16        |
| 2.8.5    | Function . . . . .                     | 17        |
| 2.9      | Limitations . . . . .                  | 22        |
| <b>3</b> | <b>Arrays</b>                          | <b>24</b> |

|          |   |           |
|----------|---|-----------|
| 3.1      | Compiling . . . . .                                 | 24        |
| 3.1.1    | Preprocessing . . . . .                             | 24        |
| 3.1.2    | Compiling . . . . .                                 | 24        |
| 3.1.3    | Assembling . . . . .                                | 24        |
| 3.1.4    | Linking . . . . .                                   | 24        |
| 3.2      | Debugging . . . . .                                 | 24        |
| 3.3      | Casting . . . . .                                   | 25        |
| 3.4      | Array . . . . .                                     | 25        |
| 3.5      | String . . . . .                                    | 26        |
| 3.6      | Command Line Arguments . . . . .                    | 34        |
| <b>4</b> | <b>Algorithms</b>                                   | <b>36</b> |
| 4.1      | Linear Search . . . . .                             | 36        |
| 4.2      | Binary Search . . . . .                             | 36        |
| 4.3      | Efficiency . . . . .                                | 37        |
| 4.3.1    | $\mathcal{O}$ Notation: . . . . .                   | 37        |
| 4.3.2    | $\Omega$ Notation: . . . . .                        | 37        |
| 4.4      | Examples . . . . .                                  | 38        |
| 4.4.1    | Linear Search . . . . .                             | 38        |
| 4.4.2    | Bad Design . . . . .                                | 39        |
| 4.4.3    | Good Design - <code>typedef struct</code> . . . . . | 40        |
| 4.5      | Bubble Sort . . . . .                               | 41        |
| 4.6      | Selection Sort . . . . .                            | 41        |
| 4.7      | Better Bubble Sort . . . . .                        | 42        |
| 4.8      | Recursion . . . . .                                 | 42        |
| 4.9      | Merge Sort . . . . .                                | 45        |
| 4.9.1    | $\Theta$ Notation . . . . .                         | 45        |
| <b>5</b> | <b>Memory</b>                                       | <b>46</b> |
| 5.1      | Hexadecimal . . . . .                               | 46        |
| 5.2      | Addresses . . . . .                                 | 46        |
| 5.2.1    | Operators . . . . .                                 | 47        |
| 5.3      | Pointers . . . . .                                  | 48        |
| 5.4      | Strings . . . . .                                   | 49        |
| 5.5      | String Comparision . . . . .                        | 51        |
| 5.6      | String Copy . . . . .                               | 53        |
| 5.7      | Malloc and Free . . . . .                           | 54        |
| 5.8      | Buffer Overflow . . . . .                           | 54        |
| 5.9      | Swap . . . . .                                      | 55        |
| 5.10     | scanf . . . . .                                     | 57        |
| 5.11     | File I/O . . . . .                                  | 58        |

|           |   |            |
|-----------|---|------------|
| <b>6</b>  | <b>Data Structures</b>                          | <b>61</b>  |
| 6.1       | Arrays . . . . .                                | 61         |
| 6.2       | Data Structures . . . . .                       | 64         |
| 6.3       | Linked List . . . . .                           | 64         |
| 6.4       | Tree . . . . .                                  | 66         |
| 6.4.1     | Binary Search Tree . . . . .                    | 66         |
| 6.5       | Hash Table . . . . .                            | 67         |
| 6.6       | Trie . . . . .                                  | 67         |
| 6.7       | Queue . . . . .                                 | 68         |
| 6.8       | Stack . . . . .                                 | 68         |
| 6.9       | Dictionary . . . . .                            | 68         |
| <b>7</b>  | <b>Python</b>                                   | <b>69</b>  |
| 7.1       | Introduction . . . . .                          | 69         |
| 7.2       | Datatypes . . . . .                             | 80         |
| 7.3       | Previous assignments from C to python . . . . . | 81         |
| 7.4       | Regular Expressions . . . . .                   | 82         |
| 7.5       | Fancier stuff: Hardware usage . . . . .         | 83         |
| <b>8</b>  | <b>Database</b>                                 | <b>86</b>  |
| 8.1       | csv files . . . . .                             | 86         |
| 8.2       | SQL . . . . .                                   | 90         |
| 8.2.1     | Example . . . . .                               | 90         |
| 8.2.2     | Relational Database . . . . .                   | 90         |
| 8.2.3     | Syntax . . . . .                                | 91         |
| 8.2.4     | Huge Database . . . . .                         | 93         |
| 8.3       | Problems . . . . .                              | 97         |
| 8.3.1     | Race Conditions . . . . .                       | 97         |
| 8.3.2     | SQL Injection Attacks . . . . .                 | 97         |
| <b>9</b>  | <b>Where to?</b>                                | <b>98</b>  |
| 9.1       | How far we have come! . . . . .                 | 98         |
| 9.2       | Tracks . . . . .                                | 99         |
| 9.2.1     | Web Programming . . . . .                       | 99         |
| 9.2.2     | Mobile App Development . . . . .                | 99         |
| 9.2.3     | Game Development . . . . .                      | 99         |
| <b>II</b> | <b>Web</b>                                      | <b>100</b> |
| <b>10</b> | <b>Introduction</b>                             | <b>101</b> |
| 10.1      | Protocols . . . . .                             | 101        |

|  |            |
|--|------------|
| 10.1.1 IP addresses . . . . .            | 101        |
| 10.1.2 Port Numbers . . . . .            | 102        |
| 10.1.3 URL: Domain Name System . . . . . | 102        |
| 10.1.4 HTTP(S) . . . . .                 | 102        |
| 10.1.5 Status Codes . . . . .            | 102        |
| <b>11 HTML</b>                           | <b>103</b> |
| <b>12 CSS</b>                            | <b>108</b> |
| <b>13 JavaScript</b>                     | <b>114</b> |
| 13.1 Syntax . . . . .                    | 114        |
| 13.2 Document Object Model . . . . .     | 115        |
| <b>14 Flask</b>                          | <b>123</b> |
| 14.1 Hello World . . . . .               | 123        |
| 14.2 Templates . . . . .                 | 124        |
| 14.3 Variables . . . . .                 | 124        |
| 14.3.1 String . . . . .                  | 124        |
| 14.3.2 Random Numbers . . . . .          | 125        |
| 14.4 Conditions . . . . .                | 126        |
| 14.4.1 Coin Flip . . . . .               | 126        |
| 14.5 Interactive Webpage . . . . .       | 127        |
| 14.5.1 Forms . . . . .                   | 127        |
| 14.6 Layouts . . . . .                   | 130        |
| 14.7 Tasks Application . . . . .         | 131        |
| <b>15 Databases</b>                      | <b>134</b> |
| 15.1 Sessions . . . . .                  | 134        |
| 15.2 SQL . . . . .                       | 135        |
| <b>III Android</b>                       | <b>138</b> |
| <b>16 Java</b>                           | <b>139</b> |
| 16.1 Introduction . . . . .              | 139        |
| 16.2 Data Types . . . . .                | 139        |
| 16.3 Examples . . . . .                  | 139        |
| 16.4 Generics . . . . .                  | 140        |
| 16.4.1 Lists . . . . .                   | 140        |
| 16.4.2 Maps . . . . .                    | 140        |
| 16.5 Classes . . . . .                   | 141        |

|                                    |            |
|------------------------------------|------------|
| 16.6 Static Methods . . . . .      | 141        |
| 16.7 Inheritance . . . . .         | 142        |
| 16.8 Interfaces . . . . .          | 142        |
| 16.9 Packages . . . . .            | 143        |
| 16.10Android . . . . .             | 143        |
| <b>17 UI</b>                       | <b>146</b> |
| 17.1 Gradle . . . . .              | 146        |
| 17.2 MVC . . . . .                 | 146        |
| 17.3 Activities . . . . .          | 146        |
| 17.4 Resources . . . . .           | 146        |
| 17.5 Layouts . . . . .             | 146        |
| 17.6 XML . . . . .                 | 146        |
| 17.7 Intent . . . . .              | 147        |
| 17.8 RecyclerView . . . . .        | 147        |
| 17.9 App Files . . . . .           | 147        |
| 17.9.1 Manifests . . . . .         | 147        |
| 17.9.2 Java Codes . . . . .        | 148        |
| 17.9.3 Layouts . . . . .           | 148        |
| 17.9.4 Values . . . . .            | 149        |
| 17.9.5 Gradle Scripts . . . . .    | 149        |
| 17.10Adding RecyclerView . . . . . | 150        |
| 17.11Adding New Activity . . . . . | 151        |

# **Part I**

## **General**

# **Part II**

## **Web**



# **Part III**

## **Android**

# Chapter 17

## UI

### 17.1 Gradle

Open Source Project - an Android Build System!

### 17.2 MVC

Model - View - Controller.

Design Pattern that advocates separating out the app in three different pieces.

### 17.3 Activities

Sort of a base class for a screen.

### 17.4 Resources

All the stuff that isn't code! Example? Layout!

### 17.5 Layouts

A layout describes how a view should look. Defined using XML.

### 17.6 XML

eXtensible Markup Language.

```

1 <LinearLayout>
2 <TextView android:text="Hello" />
3 </LinearLayout>

```

Program 17.1: sample XML code

## 17.7 Intent

Special object that represents a way to go from one activity to another.

## 17.8 RecyclerView

It basically represents anything that's a list of items.

## 17.9 App Files

From our *Pokedex* App.

### 17.9.1 Manifests

#### AndroidManifest.xml

Basically a configuration file.

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest
3   xmlns:android="http://schemas.android.com/apk/res/android"
4   package="com.example.pokedex">
5   <application
6     android:allowBackup="true"
7     android:icon="@mipmap/ic_launcher"
8     android:label="@string/app_name"
9     android:roundIcon="@mipmap/ic_launcher_round"
10    android:supportRtl="true"
11    android:theme="@style/Theme.Pokedex">
12    <activity android:name=".PokemonActivity"></activity>
13    <activity android:name=".MainActivity">
14      <intent-filter>

```

```

15         <action
            ↪ android:name="android.intent.action.MAIN"
            ↪ />
16
17         <category
            ↪ android:name="android.intent.category.LAUNCHER"
            ↪ />
18     </intent-filter>
19 </activity>
20 </application>
21
22 </manifest>

```

Program 17.2: App Files : Manifests : AndroidManifest.xml

## 17.9.2 Java Codes

Code like we've seen before.

## 17.9.3 Layouts

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <androidx.constraintlayout.widget.ConstraintLayout
    ↪ xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:app="http://schemas.android.com/apk/res-auto"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent"
7     tools:context=".MainActivity">
8
9     <androidx.recyclerview.widget.RecyclerView
10         android:layout_width="match_parent"
11         android:layout_height="match_parent"
12         android:id="@+id/recycler_view"/>
13
14 </androidx.constraintlayout.widget.ConstraintLayout>

```

Program 17.3: App Files : Layouts : activity\_main.xml

## 17.9.4 Values

### strings.xml

Helps when we want to support multiple languages.

```
1 <resources>
2     <string name="app_name">Pokedex</string>
3 </resources>
```

Program 17.4: App Files : Values : strings.xml

## 17.9.5 Gradle Scripts

```
1 plugins {
2     id 'com.android.application'
3 }
4
5 android {
6     compileSdkVersion 30
7     buildToolsVersion "30.0.3"
8
9     defaultConfig {
10         applicationId "com.example.pokedex"
11         minSdkVersion 21
12         targetSdkVersion 30
13         versionCode 1
14         versionName "1.0"
15
16         testInstrumentationRunner
17             ↪ "androidx.test.runner.AndroidJUnitRunner"
18     }
19
20     buildTypes {
21         release {
22             minifyEnabled false
23             proguardFiles
24                 ↪ getDefaultProguardFile('proguard-android-optimize.txt'),
25                 ↪ 'proguard-rules.pro'
26         }
27     }
28
29     compileOptions {
```

```

26         sourceCompatibility JavaVersion.VERSION_1_8
27         targetCompatibility JavaVersion.VERSION_1_8
28     }
29 }
30
31 dependencies {
32
33     implementation 'androidx.appcompat:appcompat:1.2.0'
34     implementation 'com.google.android.material:material:1.2.1'
35     implementation
36         ↪ 'androidx.constraintlayout:constraintlayout:2.0.4'
37     implementation 'androidx.recyclerview:recyclerview:1.1.0'
38     testImplementation 'junit:junit:4.+'
39     androidTestImplementation 'androidx.test.ext:junit:1.1.2'
40     androidTestImplementation
41         ↪ 'androidx.test.espresso:espresso-core:3.3.0'
42 }

```

Program 17.5: App Files : Gradle Scripts : build.gradle (Module app)

## 17.10 Adding Recycler View

1. Add dependency in build.grade (Module: App)
2. Start with view - What the app needs to be doing
  - (a) Add view in the layout (activity\_main.xml)
  - (b) Add ID to the view to reference
  - (c) Need a way to define how each row is going to look like
  - (d) Create new layout for that
  - (e) Add view in this layout, and IDs to reference
3. Then create models to power that view
  - (a) Create java class to represent a single element
  - (b) Add constructors, getters, and setters as per need
4. Write the controllers to hook up the two

- (a) RecyclerView class has another class attached to it called the adapter - what data is to be displayed and how to do it
- (b) Create class to represent all of the data inside the recycler view that extends `RecyclerView.Adapter`
- (c) It's a generic class that takes as its type a *ViewHolder* that holds a view and allows to manipulate what's on the screen. We're going to create an object that holds that view and from there we modify some of the layout elements we just defined
- (d) Add fields in the ViewHolder class to represent the layout and views we created
- (e) Write constructors to get views by id
- (f) Get data (or hardcode some for now)
- (g) Implement methods defined on `RecyclerView.Adapter`
  - i. `onCreateViewHolder`
  - ii. `onBindViewHolder`
  - iii. `getItemCount`

#### 5. Use the adapter

- (a) Add a few more fields in the MainActivity
  - i. `RecyclerView`
  - ii. `RecyclerView.Adapter`
  - iii. `RecyclerView.LayoutManager`
- (b) Instantiate them
- (c) Connect them

## 17.11 Adding New Activity

1. Create New Activity (right click on left hand side)
2. Start with layout
3. Next is model
4. Now *Intent*, that is how we pass data from first activity to the second
  - (a) Use `containerView.setTag` and pass the object representing the data
  - (b) Add `eventHandler`