

Basic Programming

1. The data types that are usually used in arithmetic operations are integer and float.
2. The data types that are usually used in word and character is string.
3. The output is 113.
4. The output is 2.
5. The output is 3566GoldenStreetMiami.

Flaky Test

1. Flaky test is a test which could fail or pass with the same code/configuration.
2. a. Inspect the element
 - b. DevTools will open. For finding the elements, use the Elements tab
 - c. Click the “select” tool in the upper-left corner of the DevTools pane.
 - d. Specify & write the locator query that uniquely identifies the target element
 - e. Then run the test case to test the element
3. a. Ensure that the system accepts valid CAPTCHA.
 - b. Ensure that the system rejects the invalid CAPTCHA.
 - c. Ensure that the CAPTCHA code resets as the website reloads.
 - d. Ensure that the CAPTCHA changes as the user enters wrong value.
 - e. Ensure that the code in the Image CAPTCHA is viewable.
 - f. Ensure the readability of the CAPTCHA.
 - g. Ensure that the captcha is case sensitive.
 - h. Ensure that the user receives new CAPTCHA on clicking the link "Get a new CAPTCHA".

Essay Question Topic

1. a. Automation test script using postman :

```
pm.test("Check data-type", () => {
    _.each(pm.response.json().response, (item) => {
        pm.expect(typeof item["userId"]).to.be.a(['integer']);
    })
}

pm.test("Check data-type", () => {
    _.each(pm.response.json().response, (item) => {
        pm.expect(typeof item["id"]).to.be.a(['integer']);
    })
})

pm.test("Check data-type", () => {
    _.each(pm.response.json().response, (item) => {
        pm.expect(typeof item["title"]).to.be.a(['string']);
    })
})

pm.test("Check data-type", () => {
    _.each(pm.response.json().response, (item) => {
        pm.expect(typeof item["body"]).to.be.a(['string']);
    })
})
```

Untitled Request

GET https://jsonplaceholder.typicode.com/posts/

Params Authorization Headers (7) Body Pre-request Script Tests **Code** Settings

```
15 |     pm.expect(typeof item["title"]).to.be.a(['string']);
16 | })
17 |
18 |
19 ✓ pm.test("Check data-type", () => {
20   _.each(pm.response.json().response, (item) => {
21     pm.expect(typeof item["body"]).to.be.a(['string']);
22   })
23 })
```

Test scripts are written in JavaScript, and are run after the response is received.
Learn more about tests scripts

SNIPPETS
Get an environment variable
Get a global variable
Get a variable

Body Cookies (1) Headers (25) Test Results (5/5)

Status: 200 OK Time: 239 ms Size: 27.88 KB Save Response

All Passed Skipped Failed

PASS Check data-type
PASS Check data-type
PASS Check data-type
PASS Check datatype

Activate Windows
Go to Settings to activate Windows.

b. POST

Untitled Request

POST https://jsonplaceholder.typicode.com/posts/

Params Authorization Headers (9) **Body** Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON Beautify

```
1 [
2   "title": "recommendation",
3   "body": "motorcycle",
4   "userId": 12
5 ]
```

Body Cookies (1) Headers (25) Test Results

Status: 201 Created Time: 346 ms Size: 1.17 KB Save Response

Pretty Raw Preview Visualize JSON

```
1 [
2   "title": "recommendation",
3   "body": "motorcycle",
4   "userId": 12,
5   "id": 101
6 ]
```

2. Automation test script : Add to cart using selenium

a. Selenium project test Add to cart

The screenshot shows the Selenium IDE interface with the following details:

- Project:** Add to cart
- Executing:** Add to cart
- URL:** https://www.bukalapak.com
- Script Content:**

```
Executing: Add to cart
https://www.bukalapak.com
1. open
2. set window size
3. click
4. type
5. send keys
6. mouse over
7. mouse out
8. click
9. mouse over
10. mouse out
11. click
12. type
13. send keys
14. mouseOver on linkText=Lanjut ke Pembayaran OK
15. mouseOut on linkText=Lanjut ke Pembayaran OK
16. click on css=.c-cart-dialog__cart-button OK
17. click on css=.qa-bulk-checkbox > input OK
'Add to cart' completed successfully
```
- Log:**
 - 14. mouseOver on linkText=Lanjut ke Pembayaran OK
 - 15. mouseOut on linkText=Lanjut ke Pembayaran OK
 - 16. click on css=.c-cart-dialog__cart-button OK
 - 17. click on css=.qa-bulk-checkbox > input OK
 - 'Add to cart' completed successfully
- Reference:**
 - Activate Windows 11:42:43
 - Go to Settings to activate Windows 11:42:45
 - 11:42:50

b. Homepage BukaLapak website

The screenshot shows the BukaLapak homepage with the following features:

- Header:** Situs Belanja Online dan Jual Beli, bukalapak.com, Perbarui
- Search Bar:** Coba cari Wireless Charger
- Banners:**
 - Uang Kaget Spesial Tahun Baru:** Total Hadiah Rp 100 juta. Valid from 15 Jan-14 Feb 2021.
 - SERBU SERU:** PS4 Slim Refund ke Budget Promoted Push. Valid until 31 December 2020.
- User Interface:** Rp0 Credits Rp0, Download app, BukaBantuan, Favorit, Transaksi, Pesan, Notifikasi, and a user profile icon.
- Footer:** Activate Windows, Go to Settings to activate Windows, Mulai Chat, and a red banner with icons for a smartphone, laptop, and other devices.

c. Search input

The screenshot shows the bukalapak.com website interface. A search bar at the top contains the query "Samsung S20 FE". Below the search bar is a dropdown menu listing various search suggestions related to the query. To the right of the search bar, there are several navigation links: Favorit, Transaksi, Pesan, Notifikasi, and a shopping cart icon. At the bottom of the page, there are promotional banners for a "Diskon s.d. 60%" offer and a "Hypeabis" section featuring food products.

d. Search result

The screenshot shows the search results page for "Samsung S20 FE" on bukalapak.com. The results are displayed in a grid format, showing five product listings. Each listing includes the product name, price, rating, and location. The products shown are: 1. Samsung Galaxy S20 FE - 8GB 128GB - Garansi Resmi (Rp9.129.000) 2. Samsung Galaxy S20FE 8GB 128GB S20 FE (Rp9.643.000) 3. Samsung Galaxy S20FE 8GB 128GB S20 FE 128 GB... (Rp9.605.000) 4. Samsung Galaxy S20 FE Fan Edition RAM 8GB RO... (Rp9.699.000) 5. Samsung Galaxy S20 FE 8GB 128GB Garansi Resm... (Rp9.205.000) On the left side of the results, there are filters for "FILTER POPULER" (4 ke atas, Sehari sampai, Super seller, Bekas), "Kategori Barang" (Semua Kategori: Handphone), and "Pengiriman" (Sehari sampai, Gratis ongkir, Pengiriman ke luar negeri). The right side of the page features a promotional banner for "CASHBACK UP TO 300 RIBU".

e. Product detail

The screenshot shows a product detail page for a Samsung Galaxy S20 FE - 8GB. The main image displays the phone's front and back. A dropdown menu for color is set to 'Biru'. Below the image, it says 'Stok terbatas! Tersisa 1 lagi!' (Stock is limited! Only 1 left!). Shipping information shows 'Lokasi Pelapak: Jakarta Timur' and 'Tujuan Pengiriman: Pilih lokasi untuk lihat ongkos kirim'. There are 13 couriers available. A red 'Beli Sekarang' button is prominent. The page also includes a 'Chat' button, a 'Masukkan Keranjang' button, and a note about product guarantee and payment by 17:00 WIB.

f. Add to cart

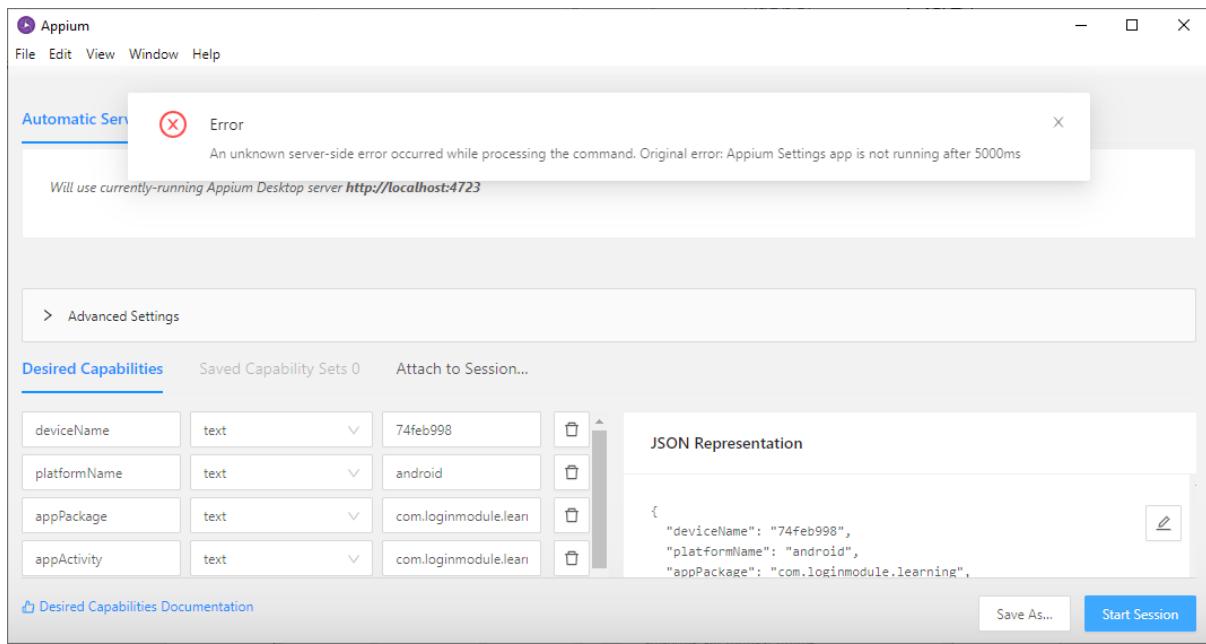
The screenshot shows a shopping cart overlay titled 'Keranjang Belanja'. It lists a single item: 'UcellCempaka Samsung Galaxy S20 FE - 8GB 128GB - Garansi Resmi' with a price of 'Rp9.129.000'. The quantity is set to 1. Below the cart, there is a 'Total Harga Barang' section showing 'Rp9.129.000' and a 'Lanjut ke Pembayaran' button. The background of the page shows the same product detail view as in the previous screenshot.

g. Cart page

The screenshot shows the bukalapak.com cart page. At the top, there's a header with the site logo, navigation links like 'Produk virtual' and 'Kategori barang', and a search bar containing 'Coba cari Kalung Masker'. Below the header is a section titled 'Keranjang belanja' (Shopping Cart). It lists a single item: 'Samsung Galaxy S20 FE - 8GB 128GB - Garansi Resmi' (Color: Biru). The item price is Rp9.129.000. To the right of the cart items, there's a summary box showing 'Total belanja Rp9.129.000' and a button labeled 'Lanjut ke Pembayaran' (Proceed to Payment). The bottom right corner of the screen displays a Windows activation message: 'Activate Windows' and 'Mulai Chat'.

3. Got some error automation test using appium not finished yet

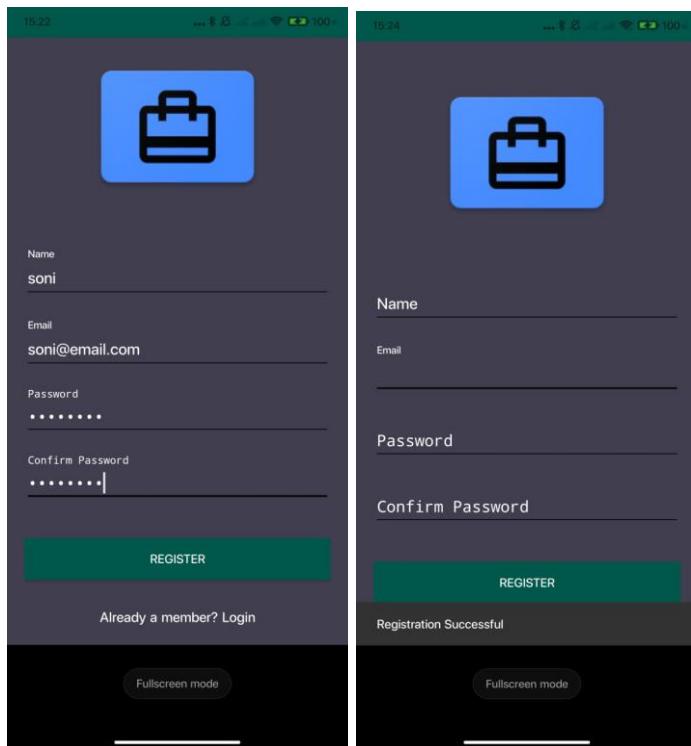
The screenshot shows the Appium desktop application interface. The main window title is 'Appium'. The 'Automatic Server' tab is selected, showing the message 'Will use currently-running Appium Desktop server <http://localhost:4723>'. Below this, there's an 'Advanced Settings' button. The 'Desired Capabilities' section is active, displaying four entries: 'deviceName' (text input field containing '74feb998'), 'platformName' (text input field containing 'android'), 'appPackage' (text input field containing 'com.loginmodule.learn'), and 'appActivity' (text input field containing 'com.loginmodule.learn'). To the right of these inputs is a 'JSON Representation' panel showing the corresponding JSON object. At the bottom of the window are buttons for 'Desired Capabilities Documentation', 'Save As...', and 'Start Session'.



a. Manual test (positive)

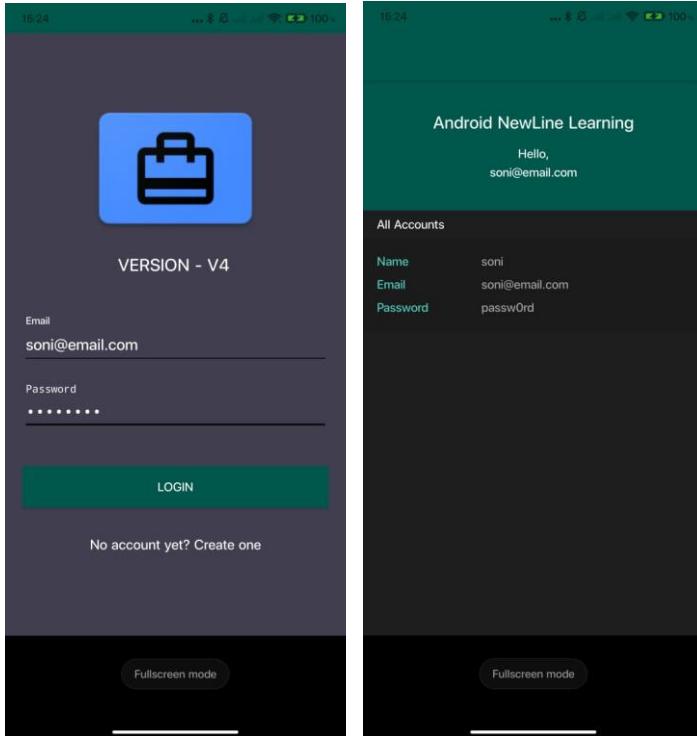
Register

Fill the register field correctly and registration successfully



Login

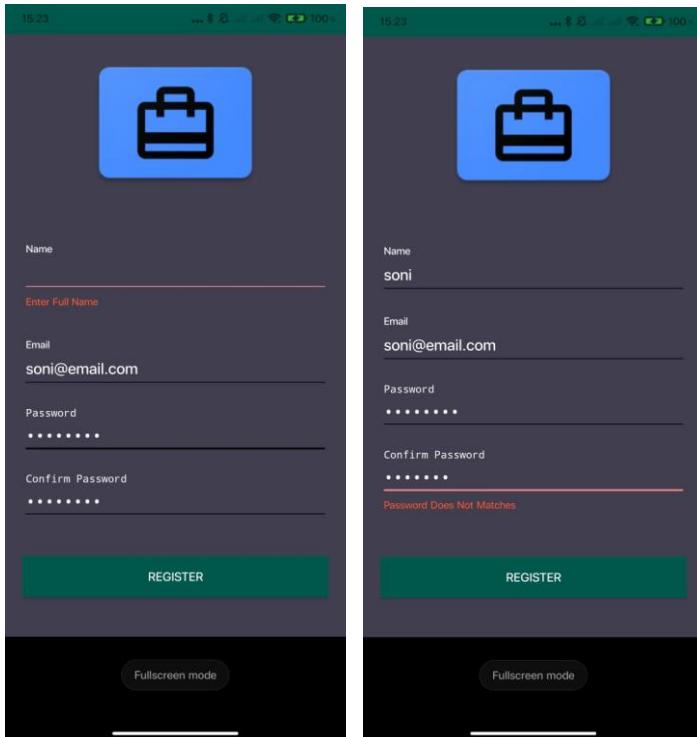
Fill the login field correctly and login successfully > go to homepage



b. Manual test (negative)

Register

Fill the register field blank and incorrect and registration unsuccessfull



Login

Fill the login field incorrect and login unsuccessfull

