Building & Testing Code

Lesson Plan

PW SKILLS

Bash scripting is a powerful tool for automating tasks in Unix-like operating systems. Here's a simple guide on how to build and test code in Bash scripting.

1. Writing a Simple Bash Script

A basic Bash script typically consists of a series of commands that you would normally run in the terminal. Here's a simple example:

```
#!/bin/bash

# A simple script to greet the user

echo "Enter your name:"
read name
echo "Hello, $name! Welcome to Bash scripting."
```

2. Making the Script Executable

Once you've written your script, you need to make it executable. This is done using the chmod command:

```
chmod +x script.sh
```

Now, you can run the script by typing:

```
./script.sh
```

3. Testing the Script

Testing in Bash can be as simple as running the script and checking its output. However, for more complex scripts, you might want to test specific functions or parts of the script.

Example of a Simple Test:

If your script has a function, you can test it like this:

```
#!/bin/bash

# A function to add two numbers
add() {
    echo $(($1 + $2))
}

# Test the add function
result=$(add 2 3)
if [ "$result" -eq 5 ]; then
    echo "Test passed!"
else
    echo "Test failed!"
fi
```

4. Debugging Bash Scripts

If something goes wrong, you can debug your Bash script by using the -x option:

```
bash -x script.sh
```

This will print each command and its result as the script runs, which helps in identifying where the script is failing.

Example of a Basic Script with Testing

```
#!/bin/bash

# A simple script to demonstrate a basic function and testing

# Function to add two numbers
add_numbers() {
  local num1=$1
  local num2=$2
  echo $((num1 + num2))
}

# Testing the function
test_add_numbers() {
  local result
  result=$(add_numbers 2 3)

  if [ "$result" -eq 5 ]; then
    echo "Test Passed"
  else
    echo "Test Failed"
  fi
}

# Run the test
test_add_numbers
```

This script includes a function to add two numbers and a basic test to check if the function works correctly. When you run the script, it will tell you if the test passed or failed.