# Homework - Chapter One

Logan Hunt

January 16, 2023

## 1 Question One

A compiler is a program that reads source in one language, and translates it to an equivalent program to another language, which can immediately be run in that language. An interpreter, on the other hand, directly executes the source as the program continues.

## 2 Question Two

### 2.1 a

Compiled machine code is typically much faster than interpreted instructions.

### 2.2 b

An interpreter can give better error diagnostics.

## 3 Question Three

In terms of portability, compiled programs are less so than interpreters. Compiled programs (to machine code) target a single architecture, and thus require seperate compilation, or cross-compilation over all target architectures. Interpreted programs on the other hand will run anywhere, as long as there is an interpreter implementation for the architecture.

## 4 Question Four

Java is a shady devil that likes to play both sides. Java source is compiled to intermediate Java bytecodes, which are then interpreted by the Java Virtual

Machine. The compilation aspect of Java here, is in the translation to JVM bytecode instructions.

# 5 Question Five

One might want to view generated assembly code to debug an issue in their code and step through the compiler's output, understand any optimizations the compiler may make, or to just explore its output.

# 6 Question Six

```
R1 = 20
R2 = 12
```

# 7 Question Seven

id2 is not a register

# 8 Question Eight

```
1. R1 = 9
2. R2 = 2
3. R1 = 11
4. R1 = 6.0
5. id3 = R1 = 6.0
```

# 9 Question Nine

## 9.1 JavaScript

- imperative
- declarative
- third-generation
- object-oriented (through prototypes)
- functional
- scripting

### 9.2 Python

- imperative
- declarative
- third-generation
- object-oriented
- functional
- scripting

## 10 Question Ten

According to the book, "A distinguishing feature of object-oriented programming is the ability of each object to invoke the appropriate method in response to a message." In C, this is not possible as structs do not have support for methods.

## 11 Question Eleven

1. Self-Hosted compilers, themselves. Compiling a compiler with an optimization bug in the hosted compiler would probably be a nightmare to fix.
2. Anything in the Linux kernel that runs in user space.
3. `malloc`

## 12 Question Twelve

This contrived program:

```
int main() {
  int i;
  for (i = 0; i < 10000000; i++);
  return 0;
}
```

It wouldn't be so bad for the compiler to just set i = 10000000.

## 13 Question Thirteen

| Declaration | Scope |
|---|---|
| `int b = 1` | $B_1$ - $B_2$ |
| `int a = 2` | $B_2$ - $B_4$ |
| `int b = 2` | $B_2$ - $B_3$ - $B_4$ |
| `int b = 3` | $B_3$ |
| `int a = 4` | $B_4$ - $B_5$ |
| `int b = 4` | $B_4$ |
| `int a = 5` | $B_5$ |

## 14 Question Fourteen

```
i = 5
j = 8
  i = 4[
j = 9
w = j - i = 9 - 4 = 5
x = j - i = 9 - 5 = 4
  j = 10
y = j - i = 10 - 5 = 5
i = 3
z = j - i = 9 - 3 = 6
```

Thus, `w = 5, x = 4, y = 5, z = 6`.

## 15 Question Fifteen

```
i = 2
j = 5
  i = 3
w = i + j = 3 + 2 = 5
x = i + j = 2 + 5 = 7
  j = 9
i = 7
y = i + j = 7 + 9 = 16
i = 6
z = i + j = 6 + 5 = 11
```

Thus, `w = 5, x = 7, y = 16, z = 11`

## 16 Question Sixteen

b : x = 4 ⇒ x = (x+3)-1 ⇒ x = 6
c : x = 1 ⇒ (x + 3) ⇒ 4
"6,4"