Practical 10:
Devasy Patel 20BCE057
Compiler Construction

Aim : To implement Code Optimization techniques: Implement any code optimization technique. The code optimization in the synthesis phase is a program transformation technique, which tries to improve the intermediate code by making it consume fewer resources (i.e. CPU, Memory) so that faster-running machine code will result. Compiler optimizing process should meet the following objectives : • The optimization must be correct, it must not, in any way, change the meaning of the program. • Optimization should increase the speed and performance of the program. • The compilation time must be kept reasonable. • The optimization process should not delay the overall compiling process.

Code:

```
import time
import math

# Loop Jamming
def loop_jamming_example():
    start = time.time()
    a = 0
    b = 0
    for i in range(5):
        a = i + 5
    for i in range(5):
        b = i + 10
    end = time.time()
    print(f"Time taken in loop jamming example: {end - start}")
```

```
def optimized_loop_jamming_example():
    start = time.time()
    a = 0
    b = 0
    for i in range(5):
        a = i + 5
        b = i + 10
    end = time.time()
    print(f"Time taken in optimized loop jamming example: {end - start}")

# Loop Invariant Method
def loop_invariant_example(x, y):
    start = time.time()
    for i in range(10):
        t = i + (x / y)
    end = time.time()
    print(f"Time taken in loop invariant example: {end - start}")

def optimized_loop_invariant_example(x, y):
    start = time.time()
    s = x / y
    for i in range(10):
        t = i + s
    end = time.time()
    print(f"Time taken in optimized loop invariant example: {end -
start}")

# Code Motion (Frequency Reduction)
def code_motion_example(x):
    start = time.time()
    i = 0
    while i < 100:
        a = math.sin(x) / math.cos(x) + i
        i += 1
    end = time.time()
    print(f"Time taken in code motion example: {end - start}")

def optimized_code_motion_example(x):
    start = time.time()
```

```
    t = math.sin(x) / math.cos(x)
    i = 0
    while i < 100:
        a = t + i
        i += 1
    end = time.time()
    print(f"Time taken in optimized code motion example: {end - start}")

# Run the examples
loop_jamming_example()
optimized_loop_jamming_example()
loop_invariant_example(10, 2)
optimized_loop_invariant_example(10, 2)
code_motion_example(1)
optimized_code_motion_example(1)
```

Output:

```
@Devasy23 ➜/workspaces/College (master) $ /home/codespace/.python/current/bin/python3 /workspaces
Time taken in loop jamming example: 3.814697265625e-06
Time taken in optimized loop jamming example: 3.0994415283203125e-06
Time taken in loop invariant example: 4.291534423828125e-06
Time taken in optimized loop invariant example: 2.384185791015625e-06
Time taken in code motion example: 7.653236389160156e-05
Time taken in optimized code motion example: 1.2874603271484375e-05
@Devasy23 ➜/workspaces/College (master) $
```