Code of detecting dead code using python :

```
import ast
def find_dead_code(source_code):
    tree = ast.parse(source_code)
    analyzer = DeadCodeAnalyzer()
    analyzer.visit(tree)
    return analyzer.dead_code

class DeadCodeAnalyzer(ast.NodeVisitor):
    def __init__(self):
        self.dead_code = set()
        self.visited_nodes = set()

    def visit_FunctionDef(self, node):
        self.visited_nodes.add(node)
        self.generic_visit(node)

    def visit_Call(self, node):
        self.visited_nodes.add(node)
        self.generic_visit(node)

    def visit_Assign(self, node):
        self.visited_nodes.add(node)
        self.generic_visit(node)

    def visit_Name(self, node):
        if isinstance(node.ctx, ast.Store) and node not in self.visited_nodes:
            self.dead_code.add(node.id)
        self.generic_visit(node)
```

```
# Example usage
source_code = """
def main():
    x = 5
    y = 10
    print(y)

if __name__ == "__main__":
    main()
"""

dead_code = find_dead_code(source_code)
print("Dead code:", dead_code)
```