

Code-Tracing & Bug-Finding Exercise Solution

1)

```
eCount = 0
oCount = 0

for x in range ( 5 ):
    if x % 2 == 0:
        print "E"
        eCount = eCount + 1
    else:
        print "O"
        oCount = oCount + 1
print "E", eCount
print "O", oCount
```

This prints:

```
E
O
E
O
E
E 3
O 2
```

2)

```
for x in [ 4,3,2,1 ]
print "x squared is ", x*x
```

This one has two problems. The header is missing ":" and the print statement must be indented.

3) Assume any value for y

```
if y > 60:
    grade = "D"
elif y > 70:
    grade = "C"
elif y > 80:
    grade = "B"
elif y > 90:
    grade = "A"
else:
    grade = "F"
```

This has a logic problem. The order forces all passing grades to be "D." The order of testing must be changed.

```
4)
    if 18 =< age and age =< 65:
        print "Working age"
```

=< is not a legal operator. It should be <=

```
5)
    if x + 17 / 3 :
        z = z + 1
    else:
        z = z - 1
```

x + 17 / 3 is not a valid condition. It evaluates to a numeric value rather than true or false.

```
6) Each time the input statement is executed, use the next value
in this list:
    5, 3, 9, 6, 4
```

```
c = 0
x = input ("enter a value")
while c < 4:
    print "value is ", x
    x = input ("enter a value")

c = c + 1
```

The problem here is an infinite loop, because c never increases within the scope of the while structure. It can be fixed by indenting c = c+1.

```
7)
def main ( ):
    x = x + 1
```

Since x = x + 1 is the first line in the program, we can tell that x has not been initialized and is therefore undefined.

```
8) x = 17
    print float (x / 2)
```

This is a logic problem. The use of the float () conversion implies that the programmer wants precision, but the integer arithmetic occurs first, so the result is 8.0

```
9) for x in range 25
    print x / 5
```

Should be "range (25)" and must also have a colon at the end of the line.

```
10)
    while x < 5:
        print "oh no!"
        x + 1 = x
```

Should be $x = x + 1$. Assignment operations require the destination to be on the left side of the operator.

```
11)
    for j in range (4, 10, 2):
        print j,
        print j * j

    print "finished"
```

Output:

```
4 16
6 36
8 64
finished
```