

# QUIZ 1 – SOLUTION

**NAME:**

## Problem 1: Variables

Which of the following variable names are legal in C++? Circle your answers. (If a name is legal but a bad choice, you should still circle it).

L **LEGAL**. However, single-character variable names should be avoided.

num\_hours **LEGAL**. This is a good, descriptive variable name.

list-of-names **ILLEGAL**. Hyphens are not allowed in variable names.

counter2 **LEGAL**. This is not too bad a variable name, though it could give rise to errors (confusing it with a variable named `counter1`)

Main **LEGAL**. However, it is a very bad name since it may easily be confused with `main`.

`__interestRate` **LEGAL**. The name itself is very good, but since C++ compilers often use double underscores, you should avoid them.

return **ILLEGAL**. This is a reserved word.

1stName **ILLEGAL**. Variable names may not start with a digit.

## Problem 2: Functions and operators

1. Write the prototype of a function named `isMultiple` that takes as arguments two integers and returns 1 if the first integer is a multiple of the second, and 0 otherwise.

```
bool isMultiple(int,int);
```

or,

```
int isMultiple(int,int);
```

2. What operator would you use and how would you use it to find whether an integer  $x$  is a multiple of another integer  $y$ ?

The `%` operator.  $x\%y$  is 0 if and only if  $x$  is a multiple of  $y$ .

## Problem 3: Scope and parameter passing

What does the following program print on the screen?

```
#include <iostream>
using namespace std;
```

```
int z = 20;
```

```
int confusion(int x, int y);
```

```
int main () {
    int x = 3, y = 5;
```

```

    int z = y / x;
    x = confusion(y, z);
    cout << "(x, y, z) = (" << x << y << z << ")\n";
    return 0;
}

int confusion (int x, int y) {
    x = x + y;
    cout << "(x, y, z) = (" << x << y << z << ")\n";
    return z;
}

```

The program prints:

```

(x,y,z) = (2051)
(x,y,z) = (6120)

```

The major points in this problem are:

- `int z = y / x;`  $z$  is now  $5/3$  which is 1 due to integer division. This is the local  $z$  (which hides the global definition of a variable with the same name).
- `confusion(y, z)` The values of the actual parameters are 5 and 1. So on entrance to `confusion`, the values of parameters  $x$  and  $y$  are 5 and 1 respectively.
- `x = x + y;`  $x$  becomes 6
- `cout ...`  $x$  and  $y$  refer to the parameters, while  $z$  is the global variable.
- `return z;` The function returns 20;
- `x = confusion(y, z)` The return value of the function, 20, is now assigned to  $x$ .
- `cout ...` Note that the `cout` statements do not print anything between the numbers.