ECM1410 'Object-Oriented Programming' CA1 - Section 3:

3.1 Explain the differences between a compiled and an interpreted programming language.

(2 marks)

A compiled programming language is one which fully translates an entire program into equivalent machine code before execution whereas an interpreted programming language translates a program line-by-line, first reading the line and then executing the command on said line.

3.2 Why Java is secure and platform-independent?

(2 marks)

Java is secure and platform-independent due to the Java Virtual Machine (JVM). The JVM is a virtual machine which Java runs on. This means that no instructions are directly executed on the computer resulting in Java being secure and able to run independently of the computer hardware.

3.3 Explain the differences and similarities between variables and constants.

(2 marks)

One similarity between variables and constants is that they both contain a value of a particular data type. A difference between them is that constants cannot be changed whereas variables can be changed during execution. In addition, constants often have a different naming scheme featuring fully capitalised names separated by underscores whereas variables are often written in camelCase.

3.4 What are the 8 primitive data types in Java? What are the differences between the types? For each type (separate lines), give a Java statement creating a variable and assigning a value.

(6 marks)

The 8 primitive data types in Java are:

- boolean
- char
- byte
- short
- int
- long
- float
- double

A 'boolean' is a data type representing 1 bit. It can either be a True or a False.

A ‘char’ is a numeric type, which means arithmetic operations can be performed on it. It can be used to represent single characters such as letters, full-stops, and special characters like line breaks.

The ‘byte’, ‘short’, ‘int’, and ‘long’ data types are integer literals. This means they represent whole numbers. The difference between them is the size allocated to each variable. A ‘byte’ is 8-bits and so can hold values from -128 to 127. A ‘short’ is 16-bits (-32,768 to 32,767), an int is a 32-bit integer (-2,147,483,648 to 2,147,483,647), and a long is 64-bits and so can hold large numbers (-2^63 to 2^63–1).

The ‘float’ and ‘double’ data types are for floating-point numbers. These are numbers which contain a decimal place. Floats can store up to 7 total digits, whereas doubles can store up to 16 digits.

```
boolean isSquare = true;

char grade = ‘A’;

byte age = 18;

short count = 200;

int year = 2021;

long seconds = 43124432;

float width = 2.5;

double ratio = 12.643;
```

3.5 What is casting? Explain the differences between implicit and explicit casting.

(2 marks)

Casting is the process of converting between data types. One difference between Implicit and Explicit casting is that implicit casting is the process of allocating more space to a variable (for example a ‘byte’ to an ‘int’) whereas explicit casting is going from larger data types to smaller data types (such as ‘long’ to ‘short’ or ‘double’ to ‘float’) losing data in the process. Explicit casting is not a precise conversion. You will often lose information. Implicit casting, on the other hand, will not produce such an issue as you are not losing any information. In addition, Implicit casting is performed automatically by the compiler, whereas Explicit casting is only done because of the programmer.

3.6 What is overflow? Please, give an example of overflow.

(2 marks)

An overflow is where you cannot store a number in an allocated primitive type due to the fact that the number is too large. For example:

```
byte a, b, c;
a = 100;
b = 50;
c = a + b;
```

The computation for the variable c would result in an overflow as you cannot store a number larger than 127 in a ‘byte’ data type.

3.7 What are the four main features of Object-Oriented Programming? Please, give a definition of each key feature.

(4 marks)

The four main features of object-oriented programming are encapsulation, abstraction, inheritance, and polymorphism.

Encapsulation, also known as data hiding, is a mechanism of wrapping the attributes and methods together as a single unit. The access modifiers of the attributes of a class are set to private, hiding them from other classes. You then provide public “setter” and “getter” methods to modify and view these attributes.

Abstraction is the hiding of all irrelevant data, leaving only what is relevant. This is done in order to reduce complexity and increase efficiency. Encapsulation is one kind of abstraction, another type being the use of abstract classes and interfaces.

Inheritance is where classes can be derived from other classes, thereby inheriting the fields and methods from those classes. This is seen often with abstract classes and with similar objects like a ‘square’ class inheriting from a ‘rectangle’ class.

Polymorphism is where a single action is performed in many different ways. Method overloading and method overriding are examples of polymorphism.