3 Concepts & Basics

3.1 Explain the differences between a compiled and an interpreted programming language. (2 marks)

A compiled programming language uses a compiler in order to translate the entire program into equivalent machine codes before carrying out any statements. This machine code is then executed. The code that is generated is completely platform dependent, meaning each compiler is designed specifically for one type of machine or architecture.

An interpreter uses an interpreter to translate the program line by line and executing after each translation. This means that the interpreter alternates between translating and executing code.

Compiled and interpreted languages are different in the way they execute the code with compiled languages translating the entire code and then executing it and the interpreter alternating between translating and executing the code.

This allows compiled languages to often be much quicker in executing code because a compiler can examine and “understand” the code and optimize as a result, reusing code within a for loop as an example or removing unnecessary bits of code.

They also differ in that in order to run code in a compiled language the code must first be compiled before it can be run, after every change in the code as well. Whereas with an interpreted language any change made to the code does not mean it has to be compiled again but it can simply be run from that same file.

3.2 Why Java is secure and platform-independent? (2 marks)

Java is secure and platform independent because it uses the Java Virtual Machine (JVM). It is a virtual machine that understands how to compile the code for different architectures. It is used in order to interpret the byte code. Because the JVM has the knowledge and ability to run on different architectures and decides on this without the user needing to give input Java can be run on different platforms, with the user not having to make any changes, making it platform independent.

Java is secure also because of the JVM. The code is compiled and executed one. Virtual machine, so if there are any potentially harmful effects of executing the code then they happen to the virtual machine and when the virtual machine shuts down, the effects with it, acting like a sandbox to execute the code in. This means that the actual machine executing the code is kept separate and therefore safe.

3.3 Explain the differences and similarities between variables and constants. (2marks)

The difference is that a constant refers to a data item that cannot be changed, whereas the data inside the named memory storage location that the variable points to can be changed.

Constants are declared similarly to variables but require the final keyword whereas variables do not use this keyword.

Constants are also named with all upper case letter and underscore between the words, whereas variables use mixed case, with the start of each word being upper case except the first word which is all lower case.

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)

1. boolean
    a. boolean boolean1 = true;
2. char
    a. char char1 = 'b';
3. byte
    a. byte byte1 = 10;
4. short
    a. short short1 = 20;
5. int
    a. int int1 = 30;
6. long
    a. long long1 = 40;
7. float
    a. float float1 = 15;
8. double
    a. double double1 = 25;

Boolean is a true or false value and does not represent a numeric value, unlike all other primitive data types. Boolean is also the only primitive data type that is 1 bit large.

The integer category covering the types char, byte, short, int, long. All of these are integer numbers unlike Booleans or floats. Integer operations can be performed on every data type in this group.

The integer subgroup of the Integer primitive data types consist of byte, short, int, long and represent an integer. Byte, short, int, long represent a different amount of bytes or memory they represent. With byte being the shortest(8 bits), next short(16 bits), next int(32 bits) and the largest amount of bytes or memory is long(64 bits). The char data type represents a single character, embedded using a unicode character in represents 16 bytes.

The float subgroup consists of float and double data types. These represent floating numbers or non integer numbers. The difference between float (32 bits) and double (64 bits) is that they represent a different number of bytes or memory space with double representing a larger number of bytes or memory space.

3.5 What is casting? Explain the differences between implicit and explicit casting. (2 marks)

Casting is when one numerical type is converted to another numerical type. Implicit casting is also known as widening and is when one converts from one numerical type to another numerical type that represents a larger number of bits. For example from a byte to a float.

Explicit casting is when a numerical type is converted to another numerical type that represents a smaller number of bits. Because the new data type may not have enough memory space to store the value and therefore java requires it to be explicitly casted. For example from a long to a short would be
long a = 20; short b = (short)a;
instead of the other way around when it is implicitly casted when short a = 20; long b = a; would suffice.

3.6 What is overflow? Please, give an example of overflow. (2 marks)

Overflow is when operations with integer type numbers produce values that are too large to be stored in the allocated data types. This will not yield an error but rather give counterintuitive values that do not represent what may have been intended. For example
byte d = (byte)130;
The variable will store -126 as the value for d instead of the intended 130.

3.7 What are the four main features of Object-Oriented Programming? Please, give a definition of each key feature. (4 marks)

1. Encapsulation
    a. Encapsulation is a mechanism of wrapping data (attributes) and code acting on the data (methods) together as a single unit. Encapsulating this is the object and the class that the object represents, keeping it private from the outside world. This is also known as data hiding since the attributes of a class are hidden from all other classes.
2. Abstraction
    a. Hiding all irrelevant data in order to reduce complexity and increase efficiency. Encapsulation is a type of abstraction. This can be achieved either through abstract classes or interfaces.
3. Inheritance
    a. Java allows classes to inherit attributes and methods from other classes by being derived from said classes. This saves a lot of repeating code
4. Polymorphism
    a. Performing a single action in different ways. There are two main ways of achieving polymorphism
        i. Compile time polymorphism is done by using method overloading
        ii. Runtime polymorphism is done by using method overriding