University of Stirling Computing Science and Mathematics

CSCU9A1

Tutorial 1 Solutions

1. What computer do you have? What is its specification, and what does it mean? As an example, consider the new MacBook Air: it has

   11.6-inch (diagonal) high-resolution LED-backlit glossy widescreen display with support for millions of colors, and for resolutions:1366 x 768 (native) at 16:9 ratio, 1152 x 720 pixels and 1024 x 640 at 16:10 ratio, and 1024 x 768 and 800 x 600 at 4:3 ratio.

   64 or 138Gbyte flash storage, 1.7GHz dual-core Intel Core i5 processor, with 3MByte shared L3 cache, 4 to 8Gbyte main memory, two USB 3 ports, and includes MacOSX 10.8.

   What does all the above mean to the naïve user? What does it mean in terms of executing instructions?

Most students will not remember or know the specs of their own machine, so use the given example instead. I usually break it down into sections as shown below and get students to explain what they understand by each section, focussing on the terms highlighted:

11.6-inch (diagonal) high-resolution LED-backlit glossy widescreen display with support for millions of colors

and for resolutions:1366 x 768 (native) at 16:9 ratio, 1152 x 720 pixels and 1024 x 640 at 16:10 ratio, and 1024 x 768 and 800 x 600 at 4:3 ratio. [Ask about reasons why one might want to use a lower resolution, e.g., to ease processor load in graphically intensive applications such as gaming; for compatibility with older software; for compatibility with different monitors,...]

64 or 138Gbyte flash storage, [Discuss the pros and cons of solid state storage over disks (SS more robust as no moving parts; disks currently cheaper).]

1.7GHz dual-core Intel Core i5 processor, [Explain that the words "core", "processor", and "CPU" are sometimes used interchangeably, but sometimes one hears of a CPU having many cores – the language can be confusing!]

with 3MByte shared L3 cache, [Don't dwell on this – they haven't yet had the lecture that explains cache. Just say it is high speed memory near the CPU that is used for storing frequently used data. Explain that this is NOT the same thing as a web browser's cache – that follows the same principle but has a very different purpose (and will be stored on the disk). In last year's exam many students did not understand this difference!]

4 to 8Gbyte main memory, [Explain that this is often also called RAM.]

two USB 3 ports, [I like to briefly describe the bad old pre-USB days of different ports for different peripherals from different manufacturers, but some tutors may be too young to recall this.]

and includes MacOSX 10.8. [Check they know what an OS is, what it does, and ask what OSs they have used.]

2. Explain the difference between using a computer program and programming a computer.

Using a computer program: using a set of instructions written by someone else (a program) to get the computer to perform some useful task for you. Programming: writing the instructions yourself.
Ask them:"When you are programming, are you using any computer programs?" (Yes – IDEs or text editors, compilers, …).

3. Discuss the benefits of using a high level language like Java over machine code.

All the benefits of a high level language: programmer can use abstract, high-level instructions, each of which can be equivalent to many detailed lines of machine code. Abstracting away from the detail makes it easier/faster to write programs and makes programs much easier to read and understand. Programs are therefore much easier to maintain and modify. Programs are also more portable, as the same program can be run on any processor provided there is a translator from the language the program is written in to the machine language of the processor.

Ask them:"Can you think of any downsides to using high-level languages?". (It may not be possible to do some things that are possible in machine code, e.g. having direct access to memory locations. Object code generated by compilers may be bigger and less efficient than hand-written machine code from an expert. In restricted execution environments, such as some embedded software, small custom-made machine code programs may be needed.)

4. What is the *Java Virtual Machine*? How does the virtual machine make Java programs more *portable* than programs written in older languages like C?

The JVM is a virtual (simulated) CPU with its own special machine code called bytecode. Implementations of the JVM exist for real CPUs; these translate bytecode instructions to instructions in the actual machine code of the target CPU. Older languages like C were compiled directly to the real machine code of a specific CPU; this meant that, once compiled, the program could only be executed by the CPU it was compiled for. Java programs are compiled into bytecode instead. The compiled bytecode can be executed on any CPU for which a JVM is available. This makes Java more portable.

5. Describe what happens when your Java program is *compiled.*

The compiler first checks that the syntax of the program is correct, reporting errors if it finds *them. If there are no errors, the compiler translates the program into bytecode. This process creates new* class files *which contain the compiled bytecode.*

6. Describe the benefits of using an IDE such as BlueJ over using a text editor and the Command Prompt to develop Java programs. Can you think of any disadvantages of using an IDE?

BlueJ is integrated, meaning that all the functions needed to write a program (editing the source code, compiling, executing the program, and many others) are all available in the same place. The programmer does not have to use many separate tools to develop the program. The BlueJ text editor has specialized features for writing programs, that are not found in general-purpose text editors. These include line numbering, syntax highlighting, and many others. IDEs also have many other advantages appreciated by advanced developers.

It's hard to think of disadvantages to using an IDE. For beginner programmers, one problem is that the IDE hides too many details so that the user does not get a full understanding of the steps that are taking place. Carrying out these steps separately outside of an IDE can give a better understanding. Learning to use the IDE itself also adds an extra learning curve for the beginner programmer to negotiate (which is why BlueJ is relatively simple compared to other IDEs). For advanced programmers, IDEs can sometimes be inflexible or difficult to configure to use the programmers preferred options, so some programmers might find it easier to just use the compiler and other tools directly.

The vast majority of programmers find that the benefits of IDEs outweigh the disadvantages.

7. What is the difference between a compile-time error and a run-time error? How many of each kind of error can you spot in the program below? Point them out.

```
public class Greeting
{
   public static void main (String[] args)
   {
       System.out.println("Hello, World!);
       System.out.print("My name is");
       System.out.prinln("R2D2");
       System.out.print("My age is");
       System.out.println(11 / (5 - 2 - 3))
   }
}
```

Compile-time error: errors in the syntax or form/structure of the code. Easily detected by the compiler when it checks the syntax. Run-time error: the program behaves in a way the programmer did not intend. Cannot be detected by the compiler so programmers must test their programs to look for these errors.

The program above has 3 compile-time errors (unclosed string, misspelled method name, missing semi-colon) and 2 run-time errors (No space between "My name is" and "R2D2", division by zero). There is a subtle difference between the two, which is worth pointing out. The first error is non-fatal:

the program does the wrong thing but it still runs. The second error is fatal: it causes the program to crash.

8. What does this program print?

```
public class Test
{
   public static void main (String[] args)
   {
      System.out.println("39 + 3");
      System.out.print(39 + 3);
   }
}
```

This is to check that they can recognize the difference between an expression and that same expression enclosed in quotes: the latter is a string. The program prints:

```
39 + 3
42
```