**Week Two:**

Part-1:

Which one of the following is an example of software development?

* Programming an app

Correct answer.

Yes - you will have to use your software development skills to program an app.

* Downloading an app
* Running an app
* Coming up with the idea for an app

Part-2:

Which one of the following is true of a **Java program**?

* Java programs can be run on any machine that has a **Java Virtual Machine** installed

Correct answer.

Yes - some machines have it pre-installed and some require the JVM to be installed.

* Java programs can **only** be run on a**PC**
* Java programs can**only** be run on a **MAC**
* None of the other answers

Part-3. Which one of the following instructions prints "Java Rocks" to the screen:

* System.out.println ("Java Rocks")
* system.out.println("Java Rocks");
* System.out.print (Java Rocks);
* System.out.print("Java Rocks");

Correct answer.

Yes - this is the correct way to print a message to the screen in Java

**Week Three**

Part-1:

Which of the following is a suitable variable declaration for a variable that can be used to hold the value of an alarm (either **true** or **false)**.

* **char**alarm;
* **double** myAlarm;
* **boolean** my alarm;
* **int**alarm;
* None of the other answers

Correct answer.

CORRECT: all other answers are wrong. The correct type is **boolean** to store true and **false**. Plus remember variable names cannot have spaces. So the following would be good for example: **boolean** alarm;

Part-2:

What is the final value of the variable 'x' after the following Java instructions are executed?

**int x = 10;**

**x = 5;**

**x = 1;**

**x = 3;**

* 1
* 3

Correct answer.

CORRECT: the final value of a variable is that last value placed in that variable. Old values are deleted.

* **19**
* 5
* 10

Part-3:

What is the output from the final **println** command below?

int i, j;

i = 7;

j = 2;

System.out.println(“total = “ + (i \* j));

* **total = i \* j**
* None of the other answers
* **total = 7 \* 2**
* **total = 14**

Correct answer.

CORRECT: the values of 'i' and 'j' ( 7 and 2 respectively) are multilied and displayed

* **19**

Part-4:

At the end of the **Part 4 video** you were shown the Java program that **JDoodle** opens up with. What is the output of that program?

\*You can see the program here:

https://www.jdoodle.com/online-java-compiler/

* None of the other answers
* **Sum of x+y = 35**

Correct answer.

CORRECT: 35 is the final value of **z** as **z** is assigned to the value of**x + y**

* **Sum of x+y = 25**
* **Sum of x+y = 10**

Part-5:

Which of the following **import**statements allows you to use the **Scanner** class in your programs for keyboard input.

* **import java.util.\*;**

Correct answer.

CORRECT: This version of the **import** statement gives access to everything in the **util** folder, including the**Scanner** class.

* **import java.Scanner;**
* **import Scanner;**
* **import java.util.Scanner;**
* None of the other answers

NO: There is more than one correct answer in the other options.

**Week Four:**

**Part-1:**

Which one of the following best describes the form of control called **selection.**

* Making choices between which instructions to execute

Correct answer.

CORRECT: yes there are 3 techniques we will look at in the following videos to implement selection.

* Jumping to a different block of instructions
* Repeating a group of instructions many times.
* None of the other answers
* Executing every instruction in the program, from the first to the last.

Part-2:

Which one of the following is the output of the final program in the part 2 video above once the user enters **20** when prompted?

* Blue
* There is an error in the program so it will crash when the user enters 20.
* Red
* Green
* None of the other answers

Correct answer.

CORRECT: The 'if' test is true (as 20 > 10) so both "Green" and "Blue" are displayed. Then the final "Red" is always displayed

* NO: There are no errors in the program.

Part-3:

Which one of the following is true of an **'if...else'**statememt

* Nested **'if...else**' statements can be be used to process **three or more**options in a program

Correct answer.

CORRECT: An **'if...else**' statements can only be be used to process **two**options in a program, but **nested** **'if...else**' statements can be used to process **more than two options**.

* None of the other answers
* An **else**statement can be used without an **if**statement.
* An **'if...else**' statement should be only be used to process **three or more**options in a program
* An **'if...else**' statement should be used to process a single choice in a program

Part-4: What is the output of the final program in the part 2 video above once the user enters **10** when prompted?

* Green
* None of the other answers
* Red

Correct answer.

CORRECT: There is no mtaching **case** and no **default,** so no instructions inisde the **switch**are executed.

* Blue

**Week-5:**

Part-1:

Which one of the following best decsribes the form of control called **iteration?**

* None of the other answers
* Jumping to a different block of instructions
* Executing every instruction in the program, from the first to the last.
* Making choices between which instructions to execute
* Repeating a group of instructions many times.

Correct answer.

CORRECT: Yes, there are 3 techniques we will look at in the following videos to implement iteration.

Part-2:

How many times does the following **for** loop execute?

**for (int i = 1; i <= 8; i = i + 2)**

{

      // code ges here

}

* This loop executes **0** times
* This loop executes **4** times

Correct answer.

CORRECT: The counter starts at 1 and goes up in steps of 2 (so, i = 1, i = 3, i = 5 and i = 7). That is 4 executions. When i becomes 9 the loop stops.

* This loop executes **5** times
* This loop executes **8** times
* None of the other answers

Part-3:

Which one of the following statements is true of a **while** loop?

* None of the other answers

Correct answer.

CORRECT: A **while**loop **can be used to repeat a group of instructions** and it is best suited to **non-fixed repetitions**. Because the test of a **while**loop is at the top, it may execute **zero or more times**.

* A **while** loop **cannot be used** to repeat a set of instructions
* A **while** loop is always the best loop to use
* A **while** loop should be used when trying to repeat a group of instructions a **fixed number of times**.
* A **while** loop executes at least once.

Part-4:

Which one of the following statements is true of a **do..while** loop?

* A **do..while** loop **cannot be used** to repeat a set of instructions
* A **do..while** loop is always the best loop to use
* None of the other answers
* A **do..while** loop executes at least once.

Correct answer.

CORRECT: As the test of the**do..while**loop is at the bottom of the loop, it will execute **at least once.**

* A **do..while** loop should be used when trying to repeat a group of instructions a **fixed number of times**.

**Week-6:**

Part-1:

Which one of the following best decsribes the concept of **structured programming?**

* Executing every instruction in the program, from the first to the last.
* Repeating a group of instructions many times.
* None of the other answers
* Making choices between which instructions to execute
* Splitting a program into a collection of blocks of code (known as **methods**) rather than just using a single (**main**) method.

Correct answer.

CORRECT:  Structured Programming is one approach to developing complex programs.

Part-2:

Consider the following line in a **main**method that **calls another method**, then pick the correct **method header**(ie top-line) for this called method from the list that follows:

**showName( );**

* **static void showNames( )**
* **static showName( )**
* **void showName( )**
* None of the other answers

Correct answer.

CORRECT: There is at least one error in all the other answers. All methods called by main should be **static**, need to have a **return type**, must have the **same name** as the method call and must end with a pair of **round brackets**. So in this case the correct method header would be

**static void showName()**

* **static void showName;**

Part-3:

Consider the following line in a **main**method that **calls another method**, then pick the correct **method header**(ie top-line) for this called method from the list that follows:

**displayResult( 12.5 );**

* **static void displayResult ( double )**
* **static void displayResult( )**
* **static void displayResult ( double numIn )**

Correct answer.

CORRECT: This method is **given an double parameter**, which matches the type of the decimal number number that was sent (12.5).

* **static void displayResult ( int numIn )**
* None of the other answers

Part-4:

Which one of the following best decsribes the concept of **method overloading**

* None of the other answers
* A s**ingle method** with **more than one loop**
* Two or more methods with the **same name** but different **parameter names**
* Two or more methods with the**same name**and the **same parameters**
* Two or more methods with the **same name** but either with different **number of parameters and/or different type of parameters**

Correct answer.

CORRECT:  Overloaded methods s**hare the same name** but can be **distinguished by different parameters**

Part-5:

Consider the following line in a **main**method that **calls another method**, then pick the correct **method header**(ie top-line) for this called method from the list that follows:

**int num = getValue ( );**

* **static int getValue ( )**

Correct answer.

CORRECT: The getName method should **return an integer value** and **receive no parameters**.

* **static void getValue()**
* **static void getValue ( int numIn )**
* **static double getValue ( )**
* None of the other answers

**Week Seven**

Part-1:

Which one of the following instructions correctly creates an array,**myNumbers**, to hold **25 integers**

* None of the other answers
* **int[ ]  myNumbers = new int [24];**
* **int[ ]  myNumbers = new int [25];**

Correct answer.

CORRECT: the type of each array element is**int**and the size is **25**.

* **double[ ]  myNumbers = new double[24];**
* **double[ ]  myNumbers = new double[25];**

Part-2:

Consider the following two lines of code:

**int i = 3;**

**int[ ] someArray = {4, 2, 9, 8, 3, 7};**

What is the value **someArray[i]?**

* **9**
* **8**

Correct answer.

CORRECT: the value of**i**is **3**, so someArray[i] is equivalent to s**omeArray[3]**. As array indexes **begin at 0**, someArray[**3**] is the **fourth**value in the list - which is **8**.

* None of the other answers
* **6**
* **5**

Part-3:

Consider a method, called **processArray**, that receives an**array of integers** as a **parameter**and **returns no value.** Which one of the following would be the **correct header**for that method?

* **static void processArray (int arrayIn)**
* None of the other answers

Correct answer.

CORRECT: All other answers have an error. The method should have a **void**return type, and **one parameter** of type**int array**. For example:

**static void processArray (int[ ] arrayIn)**

* **static void processNames (int[ ])**
* **static void processNames (arrayIn)**
* **static int[ ] processNames ( )**

Part-4:

When is it appropriate to use the **enhanced for loop**to process an array.

* It is always appropriate to use the **enhanced for loop**
* When the array has over 100 elements.
* When you just need to read the elements within an array

Correct answer.

CORRECT: The **enhanced for loop** should only be used with an array if the **array is not being modified.**

* None of the other answers
* When wishing to modify the elements within an array

Week Eight:

Part One:

Which one of these statements is true with respect to **Classes** and **Objects**

* An **object** contains just**methods**
* None of the other answers
* An **object** contains just**data**
* An **object** is a blueprint for creating **Classes**
* A **Class** is a blueprint for creating **objects**

Correct answer.

CORRECT: A class can be used to generate many objects

Part-Two:

Consider the following instruction that creates a Robot object (as discussed in the video above) then pick the one correct statement from the list that follows that moves the robot 15 positions to the right (using the **moveRight** method discussed in the video above):

**Robot myRobot = new Robot (10, 20);**

* None of the other answers
* **myRobot.moveRight(15);**

Correct answer.

CORRECT: The **moveRight**method is called by joining it to the object (**myRobot**) with a dot operator (**.**)

* **Robot.moveRight(15);**
* **moveRight(15);**
* **moveRight.myRobot(15);**

Part Three:

Consider the following instruction then pick the one correct statement from the list that follows

**Scanner kbd = new Scanner (System.in);**

* The name of the constructor is **kbd**
* None of the other answers
* The **constructor** has no parameters
* The name of the object is **Scanner**
* The name of the object is **kbd**

Correct answer.

CORRECT:The class is **Scanner**, the object is**kbd**the constructor is **Scanner( )**and this constructor has one parameter **System.in**.

Part Four:

Which one of the following instructions correctly creates a**Rectangle** object (discussed in the lecture video above), called **myRectangle**, with length 5 and height 5?

* **Rectangle myRectangle = new Rectangle (5, 5);**

Correct answer.

CORRECT: The object **myRectangle** has been created correctly with the correct class name (**Rectangle**) and parameters (**5,5**).

* **Rectangle myRectangle = new Rectangle (5);**
* None of the other answers
* **myRectangle Rectangle = new Rectangle (5, 5);**
* **Rectangle myRectangle = Rectangle (5, 5);**

Part Five:

Consider the following instruction for creating an **array of Robot objects** and assume 10 robots are added into this array, then choose the one correct instruction for moving the first robot **25** positions to the left (using the**moveLeft**method).

**Robot[] myRobot = new Robot [10];**

* **myRobot[0].moveLeft(25);**

Correct answer.

CORRECT: The first robot is **myRobots[0]** then the (**.**) dot operator can be used to call the **moveLeft**operation.

* **myRobot[0] = moveLeft(25);**
* **myRobot.moveLeft(25);**
* None of the other answers
* **myRobot[25] = moveLeft(0);**