

Section A - Multi Choice Questions.

Answer **ALL** questions in this section.

Consider the code in each question carefully, then write IN YOUR ANSWER BOOK only the question number and the letter (e.g. A, B, C or D) associated with your chosen answer.

Note: the code in these questions contains no errors.

1. Multi Choice Question (JavaScript)

2 marks

```
<script>
  var a = 9, b = 1, c = 5, d = 3;
  function Add(c, d) {
    return c * d;
  }
  alert( Add(a, b) );
</script>
```

When the code above is executed, the output is:

- | | |
|----|----|
| A) | 8 |
| B) | 15 |
| C) | 9 |
| D) | 10 |

2. Multi Choice Question (JavaScript)

2 marks

```
<script>
  var a = 5.5, b = 0.25, c = 4.5, d = 0.5;
  function Subtract(c, d) {
    return c - b;
  }
  alert( Subtract(d, c) );
</script>
```

When the code above is executed, the output is:

- | | |
|----|-------|
| A) | 4.25 |
| B) | -4 |
| C) | 4 |
| D) | 0.25 |
| E) | -0.25 |

3. Multi Choice Question (C++ v11)

3 marks

```
int a = 4, b = 3, d = 6, e = 8;

int MODIFY(int a, int b) {
    return (a - d);
}

int main() {
    int answer = MODIFY(5, 0);
    std::cout << answer;
}
```

When the code above is executed, the output is:

- | | |
|----|----|
| A) | 7 |
| B) | -2 |
| C) | -1 |
| D) | 5 |

4. Multi Choice Question (C++ v11)

4 marks

```
int Operation1(int a, int b) {
    return (a / b);
}

int Operation2(int b, int c) {
    return (b - c);
}

int main() {
    int a = 9, b = 3, c = 5;

    //don't forget to work out the nested functions first
    int RESULT = Operation2(Operation2(a,b), Operation1(b,a));
    std::cout << RESULT;
}
```

When the code above is executed, the output is:

- | | |
|----|----|
| A) | 10 |
| B) | 9 |
| C) | 8 |
| D) | 7 |
| E) | 6 |

Section B

Answer **ALL** questions in this section in our answer book.

5. **Dealing with Data, Variables and Constants**

3 marks

Explain the difference between “hard-coded” values, “variables” and “constants”.

6. **Dealing with Data, Variables and Constants**

6 marks

Please note, the following code contains no errors.

Once executed, the JavaScript function (below) correctly outputs the message “Value: 28.274333886” to the browser’s console window.

As part of your consideration of this question you are advised to “walk-through” the code much the same way as a computer would.

```
5
6  <script>
7    function calcArea(r) {
8      var PI = 3.141592654;
9      if (r === undefined) {
10         r = 3;
11      }
12      return PI * (r * r);
13    }
14    var PI = 3.141592654;
15    var v = 3;
16    console.log( "Value: " + calcArea(v) );
17  </script>
18
```

Fully explain why the output message would remain unchanged (i.e. “Value: 28.274333886”) if we modify line 15 to read:

```
var v;
```

When planning your answer - consider the design of the function, how does it deal with potentially unforeseen circumstances (does it adopt a defensive or secure style of coding), what happens in each step, does it utilise any error checking, if so how, when, would any other values work, etc?

7. **Common Overflow Programming Errors**

4 marks

Explain the cause of an “integer overflow” and how it could be avoided; support your answer by explaining the difference between a “long” and “short” int, and how this effects the values that can be stored.

8.	Common Overflow Programming Errors	4 marks
<p>Secure (and defensive) programming is generally tries to ensure that a system continues to function despite any potentially unforeseen circumstances. Explain how “out of bounds” errors (another form of “buffer overflow”) can be prevented by adopting a protective or defensive coding style that verifies sizes; support your explanation with either a graphically illustrated or described example.</p>		

9.	Programming Functions	8 marks
<p>Note: the following code contains one of more possible errors.</p> <p>During compilation, the C++ function (below) causes a number of error messages (also shown below).</p> <pre> 3 void Add2Numbers(int num1, int num2) { 4 return num1 + num2; 5 } 6 7 int main() { 8 std::cout << Add2Numbers(5, 5); 9 } </pre> <p>Compilation Error associated with the above code: <i>exit status 1</i> <i>main.cpp:4:2: error: void function 'Add2Numbers' should not return a value [-Wreturn-type]</i> <i> return num1 + num2;</i> <i> ^ ~~~~~</i></p> <p><i>main.cpp:8:13: error: invalid operands to binary expression ('std::ostream' (aka 'basic_ostream<char>') and 'void')</i> <i>std::cout << Add2Numbers(5, 5);</i> <i>~~~~~ ^</i></p>		
a.	Starting with “main.cpp”, fully explain each aspect of the second error message and how it relates to the first error.	(5 marks)
b.	After reviewing both error messages and the code you realise that you only need to change 1 thing to fix ALL the errors – Identify and explain your fix.	(3 marks)

10.	Common Types of Programming Errors	4 marks
	Explain why “compiled” programs are less likely to contain “syntax errors” when compared to “interpreted” programs; include the differences between compilers and interpreters.	

11.	Debugging <i>When writing code, it is not uncommon for programmers to make mistakes, these mistakes (bugs) need to be identified and fixed as soon as possible – an important part of programming is finding, testing and “debugging” code. The following questions are designed to give you an opportunity to demonstrate your knowledge, understanding and experience of debugging.</i>	5 marks
	In general programming terms, explain the difference between a “runtime” and “compile time” error; identify which type of error (syntax or logical) would most likely occur in each.	

12.	Programming Language Feature <i>There are many features that programmers can use when coding – the following set of questions are designed to give you the opportunity to demonstrate your knowledge, understanding and experience of programming with regards to language features.</i>	9 marks
-----	--	---------

a.	Define the term and explain why most modern programming languages are considered to be “Turing complete”.	(4 marks)
b.	<i>When answering this question consider how many things are changed on your computer, mouse positions, screen updates, etc – data is the mechanism that supports all of it.</i> As discussed in lectures, all programming languages need to be able to “organise and manipulate information” as well as being able to “express an algorithm”. What does “organise and manipulate information” mean; <u>without including sample code</u> , support your answer with an explanation of why a programmer might need to organise and manipulate “strings”.	(5 marks)

13.	IPO (Input-Process-Output) Model	6 marks
	As discussed in lectures, in terms related to “IPO” explain the purpose and advantages of “daisy chaining” functions; illustrate your explanation with a labelled, representative diagram.	

14. Flow Control

8 marks

In every procedural programming language there is a common set of logical control structures which allow programmers to control the flow of execution within a program.

a.	With regards to controlling flow within a program, describe “selection”; support your description with a small, pseudocode example illustrating this type of structure.	(4 marks)
b.	With regards to controlling flow within a program, describe “function”; support your description with a small, pseudocode example illustrating this type of structure.	(4 marks)

15. Data Declarations

9 marks

The ability to store, maintain, use and destroy data is an essential component of any program

a.	Explain the difference between a “variable” and an “array”, support your answer with an explanation of what an “index” is and how it is used with regards to array data structures.	(4 marks)
b.	Explain the difference between “local” and “global” variable scope; support your explanation with possible advantages and issues associated with the use of “global” variables.	(5 marks)

16. Program Language Operators

3 marks

In programming, operators are symbols that tell a computer what specific mathematical, relational or logical operation it needs to perform in order to produce a final result.

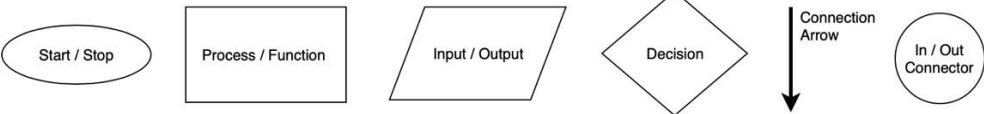
Note: to help isolate the operators, each is marked using quotes.

Briefly describe how the increment and decrement unary operators in C++, can be used to modify a value stored in a variable.

17. Algorithms

8 marks

As discussed in lectures, an algorithm is a list of instructions that are followed in order to solve a problem; algorithms need to have the right steps in the right order. Using the standard symbols (shown below) design a flowchart for each of the following problems



Using any of the appropriate standard symbols (shown above) design a flowchart for the following problem

Initialise a variable called “total” with the value 29 then create a loop that decreases “total” by 5 until “total” is less than or equal to 0.

18. Programming Fundamentals

12 marks

The following questions are setup to explore your understanding and knowledge of some of the more fundamental aspects of programming.

a.	As covered in the module, explain the difference between languages like C++, JavaScript, HTML and XML.	(4 marks)
b.	Describe the purpose and general features of an “IDE” with regards to programming.	(5 marks)
c.	In programming terms and as discussed in the module, explain the issue associated with using quotes within strings and how this issue can be mitigated using either “escaped characters” or mixed quotes; include an example of mixing quotes to avoid this issue.	(3 marks)