

COMP2212 Programming Language Concepts Coursework II

Module Code:	COMP2212		
Module Title:	PROGRAMMING LANGUAGE CONCEPTS		
Module Leader:	Muhammad Imran Babar		
Assessment Type:	Individual Coursework	Weighting:	15%
Submission Due Date:	08/04/2024 5.00 PM (Malaysia time)		
Method of Submission:	Blackboard		

This assessment relates to the following Module Learning Outcomes:

A. Knowledge and Understanding	A1. The key mechanisms underpinning the functional programming model. A2. A3.
B. Subject Specific Intellectual and Research Skills	B1. B2. B3.
C. Subject Specific Practical Skills	C1. C2. C3.

Coursework Brief:

INSTRUCTIONS:

Here is the list of the assessments that you will solve in coursework II. Your solution for the given assessment is due by the given hand in date but may be submitted at any point before that.

Repeated multiple submissions are allowed before the deadline. You will be given feedback on how well your solutions perform against our test suit. Marks will only be given to your final submission after the deadline. The coursework II comprises 15% of the overall assessment for the module.

Total Marks: 100

Weightage: 15%

ASSESSMENT A1

- a) Find the runtime error in the given code and explain the reason:

[Marks: 2]

```
using namespace std;
int main()
{
    long long N;

    long arr[N];

    cin >> N;

    for (int i = 0; i < N; i++) {
        cin >> arr[i];
    }

    for (int i = 0; i < N; i++) {
        cout << arr[i] << " ";
    }

    return 0;
}
```

- b) Define a simple grammar rule for a while loop in a programming language. [Marks: 10]
- c) Define a grammar rule for a function declaration in a programming language, including parameters and return type. [Marks: 10]
- d) A functional call is very significant in a programming language. Define a BNF grammar rule for a function call statement that supports passing parameters. [Marks: 10]

ASSESSMENT A2

- a) During lexical analysis in C++ several common tokens are generated. For the given code which tokens will be generated by the lexer. [Marks: 7]

```
#include <iostream>

int main() {
    int num1 = 42;
    double pi = 3.14159;
    if (num1 > 0) {
        std::cout << "Positive number" << std::endl;
    } else {
        std::cout << "Non-positive number" << std::endl;
    }
}
```

```
}  
return 0;  
}
```

- b) Consider the following code snippet and provide an overview of the parsing process.

[Marks: 6]

```
int main() {  
    int x = 5;  
    if (x > 0) {  
        std::cout << "Positive number" << std::endl;  
    }  
    return 0;  
}
```

- c) Consider the following C++ code snippet with nested structures and function calls:

```
#include <iostream>  
  
struct Point {  
    int x;  
    int y;  
};  
  
void printPoint(const Point& p) {  
    std::cout << "(" << p.x << ", " << p.y << ")";  
}  
  
int main() {  
    Point origin = {0, 0};  
    printPoint(origin);  
  
    if (origin.x == 0 && origin.y == 0) {  
        std::cout << "\nOrigin detected!" << std::endl;  
    }  
  
    return 0;  
}
```

Define a BNF grammar rule that encompasses the syntactic elements present in this code, including struct declaration, function definition, and if statements.

[Marks: 10]

ASSESSMENT A3

Create a parser for a simplified markup language that supports tags, attributes, and nested elements. Define a BNF grammar rule for this markup language using C++. [Marks: 15]

ASSESSMENT A4

What is Binding in C++? Explain the concepts of early and late binding in C++ and how they differ from each other. Explain with reasonable examples.

[Marks: 6]

ASSESSMENT A5

Define the concept of type inference in Haskell. Provide an example illustrating type inference.

[Marks: 6]

ASSESSMENT A6

Why is type checking important in a programming language? Explain with programming examples in any language.

[Marks: 8]

ASSESSMENT A7

Explain the concept of operational semantics in programming languages. How does it differ from denotational semantics? Explain how operational semantics rules could be defined for a simple loop statement.

[Marks: 10]

Any work submitted after the deadline's time will be subject to the standard University late penalties unless an extension has been granted, in writing by the Senior Tutor, in advance of the deadline. Details on the University's late penalties can be found here:

- <https://www.southampton.ac.uk/~assets/doc/quality-handbook/Late%20Submission.pdf>