Instructions

1. Read the question paper carefully and answer all the questions below. Write your answers in the provided answer booklet. Before you begin, make sure that you fill in all your personal details correctly and clearly on the front cover of the answer book.

2. This evaluation opportunity comprises of 3 questions on 3 pages. Please check that you have received the entire paper before you begin writing.

3. This is a closed book paper. You are not allowed to have any literature with you when writing this test.

4. Switch off your cell phone, and keep it off for the duration of the test. No electronic devices of any form are allowed to be used for the duration of the test.

5. Plagiarism or cheating of any form will not be tolerated. If you are found guilty of any such transgressions, disciplinary action will be taken. This may include suspension of your studies at the University of Pretoria.

6. Number your questions clearly and according to the numbering scheme provided in this question paper (unless otherwise instructed). Provide a clear separation between each question, preferably with a heading at the start of each new question.

7. Write clearly and concisely. Use bulleted lists to clearly organise an answer, if you are asked to discuss multiple points. If you need to continue an answer on another page, please indicate this clearly.
Question 1: Statement-Level Control Structures (17 marks)

1. The control expressions for both two-way selection statements and logically-controlled loops can take different forms in different programming languages. **Contrast** the form of control expressions in C and Java. [2]

2. Consider the following hypothetical program code example (assume that all variables have been declared, and that the program syntax is correct):

   ```
   if (sum == 0)
     if (count == 0)
       print("Output 1");
     else
       print("Output 2");
   ```

   The above program code example is considered ambiguous. Answer the following questions:
   (a) **Explain** why the above example is ambiguous. [2]
   (b) **Explain** how Python eliminates the problem of ambiguity described above. [1]
   (c) **Explain** how Ruby eliminates the problem of ambiguity described above. [1]

3. Consider multiple-way selection statements, and answer the following questions:
   (a) C and C++ multiple-way selection statements allow control to flow through more than one segment (i.e. case label). **Explain** how C# handles control flow through more than one segment. [2]
   (b) Ada multiple selection statements may be considered more reliable than C multiple selection statements. **Explain** why this is the case. [1]

4. Consider counter-controlled loops and logically-controlled loops, and answer the following questions:
   (a) Can every counter-controlled loop be written as a logically-controlled loop? [1]
   (b) Can every logically-controlled loop be written as a counter-controlled loop? [1]

5. **Contrast** the semantics of a logically-controlled pretest loop and the semantics of a logically controlled posttest loop. [2]

6. Consider the concept of unconditional branching, and answer the following questions:
   (a) **Explain** the purpose of an unconditional branch statement. [1]
   (b) **Explain** Java’s approach to unconditional branching. [1]

7. Consider the following example of a hypothetical multiple selection using if:

   ```
   if (condition_1) : ...
   elif (condition_2) : ...
   elif (condition_3) : ...
   ```

   In contrast, also consider the following example of Dijkstra’s selection guarded statement:

   ```
   if condition_1 -> ... 
   [] condition_2 -> ... 
   [] condition_3 -> ... 
   ```

   **Contrast** between the semantics of each of these two examples, in a situation in which more than one condition is satisfied at the same time.

Question 2: Subprograms (17 marks)

1. There are two major types of abstraction that are provided for in modern programming languages. **Name** the type of abstraction that is introduced by means of subprograms. [1]

2. **Differentiate** between formal parameters and actual parameters. Make use of a program code example if this will make your explanation easier. [2]

3. **Differentiate** between functions and procedures. Make use of a program code example if this will make your explanation easier. [4]
4. Consider the concept of pass-by-reference parameters, and answer the following questions:
   (a) **Describe** one (1) advantage of pass-by-reference parameters. [1]
   (b) **Describe** one (1) disadvantage of pass-by-reference parameters. [1]

5. Consider the following hypothetical program code example, written in a language that supports parameters that are subprogram names:

```plaintext
function main() {
    var a = 1;

    function sub1() {
        var a = 2;
        call sub2(sub3);
    }

    function sub2(sub_param) {
        var a = 3;
        call sub_param;
    }

    function sub3() {
        print(a);
    }

    sub1();
}
```

Assume static scoping, and that the call `print(x)` outputs the value of `x`. Answer the following questions:
   (a) **Give the output** of the program code example, if shallow binding is assumed. [1]
   (b) **Give the output** of the program code example, if deep binding is assumed. [1]

6. **Contrast** the approaches used by C++ and Ada to disambiguate overloaded subprograms. [2]

7. Consider the concept of generic subprograms, and answer the following questions:
   (a) **Explain** how the compiler handles a call to a generic subprogram in C++. [2]
   (b) **Explain** how the compiler handles a call to a generic subprogram in Java. [2]

**Question 3: Abstract Data Types (16 marks)**

1. There are two major types of abstraction that are provided for in modern programming languages. **Name** the type of abstraction that is introduced by means of abstract data types and object-oriented programming. [1]

2. **Describe** the two conditions that an abstract data type must satisfy. [2]

3. **Give a practical example** of a situation in which a friend function should be used in a C++ program. [1]

4. Object variables in the C# programming language are always allocated dynamically from the heap. **Explain** how heap storage is usually reclaimed in C#. [1]

5. Classes in the Ruby programming language are considered dynamic. **Explain** what a dynamic class is, in the context of Ruby. [2]

6. Consider the concept of an encapsulation construct (not a naming encapsulation), and answer the following questions:
   (a) **Describe** the two (2) special needs that encapsulation constructs fulfil in the context of large programs. [2]
   (b) **Explain** how encapsulation is achieved in the C programming language. [3]

7. Consider the concept of naming encapsulations, and answer the following questions:
   (a) **Explain** the purpose of naming encapsulations, in terms of the problem they solve, and how they achieve this solution. [2]
   (b) **Give the name** of the naming encapsulation construct that is used in C++. [1]
   (c) **Give the name** of the naming encapsulation construct that is used in Java. [1]