**Dr. D. Y. Patil Institute of Technology, Pimpri, Pune**

**Department of Computer Engineering**

**Subject-PPL(210255)**

**MCQ- Unit-1 Fundamentals of Programming**

|  |  |
| --- | --- |
| ((MARKS)) (1/2/3...) | 1 |
| ((QUESTION)) | The feature by which the program that work on one platform can be modified on other platform is called ------------------ |
| ((OPTION\_A)) | Reusability |
| ((OPTION\_B)) | Portability |
| ((OPTION\_C)) | Robustness |
| ((OPTION\_D)) | Locality |
| ((CORRECT\_CHOICE)) (A/B/C/D) | B |
| ((EXPLANATION)) (OPTIONAL) |  |

|  |  |
| --- | --- |
| ((MARKS)) (1/2/3...) | 1 |
| ((QUESTION)) | Semantic of a program means— |
| ((OPTION\_A)) | Format of a program |
| ((OPTION\_B)) | Meaning of a program |
| ((OPTION\_C)) | Simply content of a program |
| ((OPTION\_D)) | None of these |
| ((CORRECT\_CHOICE)) (A/B/C/D) | B |
| ((EXPLANATION)) (OPTIONAL) |  |

|  |  |
| --- | --- |
| ((MARKS)) (1/2/3...) | 1 |
| ((QUESTION)) | A program which convert high level programming language into machine code is called-------- |
| ((OPTION\_A)) | Translator |
| ((OPTION\_B)) | Compiler |
| ((OPTION\_C)) | Assembler |
| ((OPTION\_D)) | None of these |
| ((CORRECT\_CHOICE)) (A/B/C/D) | B |
| ((EXPLANATION)) (OPTIONAL) |  |

|  |  |
| --- | --- |
| ((MARKS)) (1/2/3...) | 1 |
| ((QUESTION)) | Which was the first language for scientific applications? |
| ((OPTION\_A)) | ALGOL60 |
| ((OPTION\_B)) | FORTRAN |
| ((OPTION\_C)) | LISP |
| ((OPTION\_D)) | COBOL |
| ((CORRECT\_CHOICE)) (A/B/C/D) | B |
| ((EXPLANATION)) (OPTIONAL) |  |

|  |  |
| --- | --- |
| ((MARKS)) (1/2/3...) | 1 |
| ((QUESTION)) | Which was the first high level language developed for business purpose? |
| ((OPTION\_A)) | ALGOL 60 |
| ((OPTION\_B)) | LISP |
| ((OPTION\_C)) | COBOL |
| ((OPTION\_D)) | FORTRAN |
| ((CORRECT\_CHOICE)) (A/B/C/D) | C |
| ((EXPLANATION)) (OPTIONAL) |  |

|  |  |
| --- | --- |
| ((MARKS)) (1/2/3...) | 1 |
| ((QUESTION)) | Which was the first language for Artificial intelligence? |
| ((OPTION\_A)) | ALGOL60 |
| ((OPTION\_B)) | FORTRAN |
| ((OPTION\_C)) | LISP |
| ((OPTION\_D)) | COBOL |
| ((CORRECT\_CHOICE)) (A/B/C/D) | C |
| ((EXPLANATION)) (OPTIONAL) |  |

|  |  |
| --- | --- |
| ((MARKS)) (1/2/3...) | 1 |
| ((QUESTION)) | Language which support one particular paradigm is: |
| ((OPTION\_A)) | Smalltalk |
| ((OPTION\_B)) | C |
| ((OPTION\_C)) | Java |
| ((OPTION\_D)) | Perl |
| ((CORRECT\_CHOICE)) (A/B/C/D) | A |
| ((EXPLANATION)) (OPTIONAL) |  |

|  |  |
| --- | --- |
| ((MARKS)) (1/2/3...) | 1 |
| ((QUESTION)) | From the point of view of the programmer what are the major advantages of using a high-level language rather than internal machine code or assembler language? |
| ((OPTION\_A)) | Program portability |
| ((OPTION\_B)) | **Easy development** |
| ((OPTION\_C)) | Efficiency |
| ((OPTION\_D)) | Portability |
| ((CORRECT\_CHOICE)) (A/B/C/D) | B |
| ((EXPLANATION)) (OPTIONAL) |  |

|  |  |
| --- | --- |
| ((MARKS)) (1/2/3...) | 1 |
| ((QUESTION)) | Which of the following is the functionality of ‘Data Abstraction’? |
| ((OPTION\_A)) | Reduce Complexity |
| ((OPTION\_B)) | Binds together code and data |
| ((OPTION\_C)) | Parallelism |
| ((OPTION\_D)) | None of the above |
| ((CORRECT\_CHOICE)) (A/B/C/D) | A |
| ((EXPLANATION)) (OPTIONAL) |  |

|  |  |
| --- | --- |
| ((MARKS)) (1/2/3...) | 1 |
| ((QUESTION)) | Which of the following supports the concept of hierarchical classification? |
| ((OPTION\_A)) | Polymorphism |
| ((OPTION\_B)) | Encapsulation |
| ((OPTION\_C)) | Abstraction |
| ((OPTION\_D)) | Inheritance |
| ((CORRECT\_CHOICE)) (A/B/C/D) | A |
| ((EXPLANATION)) (OPTIONAL) |  |

|  |  |
| --- | --- |
| ((MARKS)) (1/2/3...) | 1 |
| ((QUESTION)) | What is printed by the print statements in the program P1 assuming call by reference parameter passing?  Program Pl()  {  x=10;  y=3;  func1(y, x, x);  print x;  print y;  }  func1 (x, y, z)  {  y = y + 4;  z = x + y + z;  } |
| ((OPTION\_A)) | 10, 3 |
| ((OPTION\_B)) | 31, 3 |
| ((OPTION\_C)) | 27, 7 |
| ((OPTION\_D)) | None of the above |
| ((CORRECT\_CHOICE)) (A/B/C/D) | B |
| ((EXPLANATION)) (OPTIONAL) | Here, we are passing the variables by call by reference. This means that the changes that we will make in the parameter would be reflected in the passed argument.  Here, the first variable passed in the function func1 (i.e., y) points to the address of the variable x.  Similarly, the second variable passed in the function func1 (i.e., x) points to the address of the variable y and the third variable passed in the function func1 (i.e., x) points to the address of the variable z. |

|  |  |
| --- | --- |
| ((MARKS)) (1/2/3...) | 1 |
| ((QUESTION)) | Which of the these is the functionality of ‘Encapsulation’? |
| ((OPTION\_A)) | Binds together code and data |
| ((OPTION\_B)) | Using single interface for general class of actions. |
| ((OPTION\_C)) | Reduce Complexity |
| ((OPTION\_D)) | All of the mentioned |
| ((CORRECT\_CHOICE)) (A/B/C/D) | A |
| ((EXPLANATION)) (OPTIONAL) |  |

|  |  |
| --- | --- |
| ((MARKS)) (1/2/3...) | 1 |
| ((QUESTION)) | Which of the following mechanisms is/are provided by Object Oriented Language to implement Object Oriented Model? |
| ((OPTION\_A)) | Encapsulation |
| ((OPTION\_B)) | Inheritance |
| ((OPTION\_C)) | Polymorphism |
| ((OPTION\_D)) | All of the mentioned |
| ((CORRECT\_CHOICE)) (A/B/C/D) | D |
| ((EXPLANATION)) (OPTIONAL) |  |

|  |  |
| --- | --- |
| ((MARKS)) (1/2/3...) | 1 |
| ((QUESTION)) | Consider the following program  Program P2  var n: int:  procedure W(var x: int)  begin  x=x+1;  print x;  end  procedure D  begin  var n: int;  n=3;  W(n);  end  begin //beginP2  n=10;  D;  End  If the language has dynamic scoping and parameters are passed by reference, what will be printed by the program? |
| ((OPTION\_A)) | 10 |
| ((OPTION\_B)) | 11 |
| ((OPTION\_C)) | 3 |
| ((OPTION\_D)) | None of the above |
| ((CORRECT\_CHOICE)) (A/B/C/D) | D |
| ((EXPLANATION)) (OPTIONAL) | In static scoping or compile-time scoping the free variables (variables used in a function that are neither local variables nor parameters of that function) are referred as global variables because at compile only global variables are available.  In dynamic scoping or run-time scoping the free variables are referred as the variables in the most recent frame of function call stack. In the given code in the function call of procedure W the local variable x is printed i.e 4. Under dynamic scoping if x would have not been there in procedure W then we would refer to x of the function in function call stack i.e procedure D and the main function but since x is a local variable not a free variable we referred to the local variable hence 4 will be printed. |

|  |  |
| --- | --- |
| ((MARKS)) (1/2/3...) | 1 |
| ((QUESTION)) | In high level programming language Pascal, each program statement ends with the |
| ((OPTION\_A)) | comma |
| ((OPTION\_B)) | semicolon |
| ((OPTION\_C)) | double quotation marks |
| ((OPTION\_D)) | single quotation marks |
| ((CORRECT\_CHOICE)) (A/B/C/D) | B |
| ((EXPLANATION)) (OPTIONAL) |  |

|  |  |
| --- | --- |
| ((MARKS)) (1/2/3...) | 1 |
| ((QUESTION)) | Importance of Studying Programming Languages is/are: |
| ((OPTION\_A)) | Increased capacity to express ideas |
| ((OPTION\_B)) | Improved background for choosing appropriate languages |
| ((OPTION\_C)) | Overall advancement of computing |
| ((OPTION\_D)) | All of this |
| ((CORRECT\_CHOICE)) (A/B/C/D) | D |
| ((EXPLANATION)) (OPTIONAL) |  |

|  |  |
| --- | --- |
| ((MARKS)) (1/2/3...) | 1 |
| ((QUESTION)) | language have adopted which features that aid separate compilation: |
| ((OPTION\_A)) | Extern |
| ((OPTION\_B)) | Breaking point features |
| ((OPTION\_C)) | Execution trace feature |
| ((OPTION\_D)) | None of the above |
| ((CORRECT\_CHOICE)) (A/B/C/D) | A |
| ((EXPLANATION)) (OPTIONAL) |  |

|  |  |
| --- | --- |
| ((MARKS)) (1/2/3...) | 1 |
| ((QUESTION)) | Which of the following is not an OOPS concept? |
| ((OPTION\_A)) | Encapsulation |
| ((OPTION\_B)) | Polymorphism |
| ((OPTION\_C)) | Exception |
| ((OPTION\_D)) | Abstraction |
| ((CORRECT\_CHOICE)) (A/B/C/D) | C |
| ((EXPLANATION)) (OPTIONAL) |  |

|  |  |
| --- | --- |
| ((MARKS)) (1/2/3...) | 1 |
| ((QUESTION)) | Function is basic building blocks in which language: |
| ((OPTION\_A)) | Imperative Languages |
| ((OPTION\_B)) | Applicative Languages |
| ((OPTION\_C)) | Rule-based Languages |
| ((OPTION\_D)) | Object-oriented programming |
| ((CORRECT\_CHOICE)) (A/B/C/D) | B |
| ((EXPLANATION)) (OPTIONAL) |  |

|  |  |
| --- | --- |
| ((MARKS)) (1/2/3...) | 1 |
| ((QUESTION)) | In testing/ debugging assertion means: |
| ((OPTION\_A)) | Control expression |
| ((OPTION\_B)) | Conditional expression |
| ((OPTION\_C)) | Both a & b |
| ((OPTION\_D)) | None of the above |
| ((CORRECT\_CHOICE)) (A/B/C/D) | B |
| ((EXPLANATION)) (OPTIONAL) |  |

|  |  |
| --- | --- |
| ((MARKS)) (1/2/3...) | 1 |
| ((QUESTION)) | What does the following fragment of C-program print?  char c[] = "GATE2021";  char \*p =c;  printf("%s", p + p[3] - p[1]) ; |
| ((OPTION\_A)) | GATE2021 |
| ((OPTION\_B)) | E2021 |
| ((OPTION\_C)) | 2021 |
| ((OPTION\_D)) | 021 |
| ((CORRECT\_CHOICE)) (A/B/C/D) | B |
| ((EXPLANATION)) (OPTIONAL) |  |

|  |  |
| --- | --- |
| ((MARKS)) (1/2/3...) | 1 |
| ((QUESTION)) | When a function calls itself, it is known as………. |
| ((OPTION\_A)) | Self Referential |
| ((OPTION\_B)) | Recursion |
| ((OPTION\_C)) | Repeated Call |
| ((OPTION\_D)) | Loop |
| ((CORRECT\_CHOICE)) (A/B/C/D) | B |
| ((EXPLANATION)) (OPTIONAL) |  |

|  |  |
| --- | --- |
| ((MARKS)) (1/2/3...) | 1 |
| ((QUESTION)) | Which these does not represent object oriented design ? |
| ((OPTION\_A)) | It follows regular procedural decomposition in favor of class and object decomposition |
| ((OPTION\_B)) | Programs are thought of collection of objects |
| ((OPTION\_C)) | Central model represents class diagrams that show the classes comprising a program and their relationships to one another |
| ((OPTION\_D)) | Object-oriented methods incorporates Structural methods |
| ((CORRECT\_CHOICE)) (A/B/C/D) | A |
| ((EXPLANATION)) (OPTIONAL) | It does not follow regular procedural decomposition. |

|  |  |
| --- | --- |
| ((MARKS)) (1/2/3...) | 1 |
| ((QUESTION)) | The feature of the object oriented paradigm which helps code  Reuse |
| ((OPTION\_A)) | object |
| ((OPTION\_B)) | class |
| ((OPTION\_C)) | inheritance. |
| ((OPTION\_D)) | aggregation |
| ((CORRECT\_CHOICE)) (A/B/C/D) | C |
| ((EXPLANATION)) (OPTIONAL) |  |

|  |  |
| --- | --- |
| ((MARKS)) (1/2/3...) | 1 |
| ((QUESTION)) | An object encapsulates |
| ((OPTION\_A)) | Data |
| ((OPTION\_B)) | Behaviour |
| ((OPTION\_C)) | State |
| ((OPTION\_D)) | Both Data and behaviour |
| ((CORRECT\_CHOICE)) (A/B/C/D) | D |
| ((EXPLANATION)) (OPTIONAL) |  |

|  |  |
| --- | --- |
| ((MARKS)) (1/2/3…) | 1 |
| ((QUESTION)) | Which of the following is not the member of class? |
| ((OPTION\_A)) | Static function |
| ((OPTION\_B)) | Virtual function |
| ((OPTION\_C)) | Friend function |
| ((OPTION\_D)) | Constant function |
| ((CORRECT\_CHOICE))(A/B/C/D) | A |

|  |  |
| --- | --- |
| ((MARKS)) (1/2/3…) | 1 |
| ((QUESTION)) | In Object-oriented programming, the problem is divided into : |
| ((OPTION\_A)) | Classes |
| ((OPTION\_B)) | Objects |
| ((OPTION\_C)) | Functions |
| ((OPTION\_D)) | Structures |
| ((CORRECT\_CHOICE))(A/B/C/D) | A |

|  |  |
| --- | --- |
| ((MARKS)) (1/2/3…) | 1 |
| ((QUESTION)) | Which of the following term is used for a function defined inside a class? |
| ((OPTION\_A)) | Member variable |
| ((OPTION\_B)) | Member function |
| ((OPTION\_C)) | Class function |
| ((OPTION\_D)) | Friend functions |
| ((CORRECT\_CHOICE))(A/B/C/D) | B |

|  |  |
| --- | --- |
| ((MARKS)) (1/2/3…) | 1 |
| ((QUESTION)) | A class is \_\_\_\_ datatype |
| ((OPTION\_A)) | Primitive |
| ((OPTION\_B)) | Derived |
| ((OPTION\_C)) | User derived |
| ((OPTION\_D)) | All above |
| ((CORRECT\_CHOICE))(A/B/C/D) | C |

|  |  |
| --- | --- |
| ((MARKS)) (1/2/3…) | 1 |
| ((QUESTION)) | A \_\_\_\_\_\_\_\_\_\_ is a special method used to initialize the instance variable of a class. |
| ((OPTION\_A)) | Member function |
| ((OPTION\_B)) | Constructor |
| ((OPTION\_C)) | Destructor |
| ((OPTION\_D)) | Structure |
| ((CORRECT\_CHOICE))(A/B/C/D) | B |

|  |  |
| --- | --- |
| ((MARKS)) (1/2/3…) | 1 |
| ((QUESTION)) | Creating a new class using one or more existing classes is known as \_\_\_\_. |
| ((OPTION\_A)) | Overloading |
| ((OPTION\_B)) | Polymorphism |
| ((OPTION\_C)) | Encapsulation |
| ((OPTION\_D)) | Inheritance |
| ((CORRECT\_CHOICE))(A/B/C/D) | D |

|  |  |
| --- | --- |
| ((MARKS)) (1/2/3…) | 1 |
| ((QUESTION)) | Ability of an operator or function call to take different forms is known as \_\_\_\_. |
| ((OPTION\_A)) | Overloading |
| ((OPTION\_B)) | Polymorphism |
| ((OPTION\_C)) | Encapsulation |
| ((OPTION\_D)) | Inheritance |
| ((CORRECT\_CHOICE))(A/B/C/D) | B |

|  |  |
| --- | --- |
| ((MARKS)) (1/2/3…) | 1 |
| ((QUESTION)) | Which one of the following is correct about the statements given below?  I: All function calls are resolved at compile-time in Procedure Oriented Programming.  II: All function calls are resolved at compile-time in OOPS. |
| ((OPTION\_A)) | I |
| ((OPTION\_B)) | II |
| ((OPTION\_C)) | ONLY II |
| ((OPTION\_D)) | ONLY I |
| ((CORRECT\_CHOICE))(A/B/C/D) | D |

|  |  |
| --- | --- |
| ((MARKS)) (1/2/3…) | 1 |
| ((QUESTION)) | Which of the following concepts means wrapping up of data and functions together? |
| ((OPTION\_A)) | Abstraction |
| ((OPTION\_B)) | Encapsulation |
| ((OPTION\_C)) | Inheritance |
| ((OPTION\_D)) | Polymorphism |
| ((CORRECT\_CHOICE))(A/B/C/D) | B |

----------------------------------------

|  |  |
| --- | --- |
| ((MARKS)) (1/2/3…) | 1 |
| ((QUESTION)) | The major components of Computer that are associated with programming language are: |
| ((OPTION\_A)) | Primitive Operations |
| ((OPTION\_B)) | Sequence Control |
| ((OPTION\_C)) | Data access |
| ((OPTION\_D)) | Storage management |
| ((OPTION\_E)) | All of the above |
| ((CORRECT\_CHOICE))(A/B/C/D) | E |

|  |  |
| --- | --- |
| ((MARKS)) (1/2/3...) | 1 |
| ((QUESTION)) | A small high-speed data storage that is between main memory  and the central processor is --------------- |
| ((OPTION\_A)) | high-speed registers |
| ((OPTION\_B)) | external files |
| ((OPTION\_C)) | Cache memory |
| ((OPTION\_D)) | Primary memory |
| ((CORRECT\_CHOICE)) (A/B/C/D) | C |
| ((EXPLANATION)) (OPTIONAL) |  |

|  |  |
| --- | --- |
| ((MARKS)) (1/2/3...) | 1 |
| ((QUESTION)) | Firmware A set of machine-language instructions implemented  by programs, called ----------- |
| ((OPTION\_A)) | microprograms |
| ((OPTION\_B)) | subprogram |
| ((OPTION\_C)) | software |
| ((OPTION\_D)) | None of the above |
| ((CORRECT\_CHOICE)) (A/B/C/D) | A |
| ((EXPLANATION)) (OPTIONAL) |  |

|  |  |
| --- | --- |
| ((MARKS)) (1/2/3...) | 1 |
| ((QUESTION)) | Binding Occurs at------- |
| ((OPTION\_A)) | language definition |
| ((OPTION\_B)) | language implementation |
| ((OPTION\_C)) | None of the above |
| ((OPTION\_D)) | Both A & B |
| ((CORRECT\_CHOICE)) (A/B/C/D) | D |
| ((EXPLANATION)) (OPTIONAL) |  |

|  |  |
| --- | --- |
| ((MARKS)) (1/2/3...) | 1 |
| ((QUESTION)) | Does the compiler program translate the whole source code in  one step? |
| ((OPTION\_A)) | No |
| ((OPTION\_B)) | Depends on the Compiler |
| ((OPTION\_C)) | Don't Know |
| ((OPTION\_D)) | Yes |
| ((CORRECT\_CHOICE)) (A/B/C/D) | d |
| ((EXPLANATION)) (OPTIONAL) | The compiler is that program which translates the whole high-level code into the machine code at once. |

|  |  |
| --- | --- |
| ((MARKS)) (1/2/3...) | 1 |
| ((QUESTION)) | #include is called |
| ((OPTION\_A)) | Preprocessor directive |
| ((OPTION\_B)) | Inclusion directive |
| ((OPTION\_C)) | File inclusion directive |
| ((OPTION\_D)) | None of the mentioned |
| ((CORRECT\_CHOICE)) (A/B/C/D) | A |
| ((EXPLANATION)) (OPTIONAL) |  |

|  |  |
| --- | --- |
| ((MARKS)) (1/2/3...) | 1 |
| ((QUESTION)) | Which of the following is the functionality of ‘Data  Abstraction’? |
| ((OPTION\_A)) | Reduce Complexity (b) Binds together code and data  (c) Parallelism (d) None of the mentioned  Reduce Complexity (b) Binds together code and data  (c) Parallelism (d) None of the mentioned  Reduce Complexity |
| ((OPTION\_B)) | Binds together code and data |
| ((OPTION\_C)) | Parallelism |
| ((OPTION\_D)) | None of the mentioned |
| ((CORRECT\_CHOICE)) (A/B/C/D) | A |
| ((EXPLANATION)) (OPTIONAL) |  |

|  |  |
| --- | --- |
| ((MARKS)) (1/2/3...) | 1 |
| ((QUESTION)) | The following are programming paradigms: |
| ((OPTION\_A)) | Procedural, object-imperative, scripting, declaring, functional, aspect-oriented |
| ((OPTION\_B)) | High Level Language, Low Level Language |
| ((OPTION\_C)) | Java, C++, Pascal |
| ((OPTION\_D)) | Procedural, object-oriented , Logic |
| ((CORRECT\_CHOICE)) (A/B/C/D) | D |
| ((EXPLANATION)) (OPTIONAL) |  |

|  |  |
| --- | --- |
| ((MARKS)) (1/2/3...) | 1 |
| ((QUESTION)) | This paradigm tells how to solve something rather than what to  solve: |
| ((OPTION\_A)) | object-oriented |
| ((OPTION\_B)) | procedural |
| ((OPTION\_C)) | scripting |
| ((OPTION\_D)) | functional |
| ((CORRECT\_CHOICE)) (A/B/C/D) | B |
| ((EXPLANATION)) (OPTIONAL) |  |

|  |  |
| --- | --- |
| ((MARKS)) (1/2/3...) | 1 |
| ((QUESTION)) | Which of these is the functionality of ‘Encapsulation’? |
| ((OPTION\_A)) | Binds together code and data |
| ((OPTION\_B)) | Using single interface for general class of actions. |
| ((OPTION\_C)) | Reduce Complexity |
| ((OPTION\_D)) | All of the mentioned |
| ((CORRECT\_CHOICE)) (A/B/C/D) | A |
| ((EXPLANATION)) (OPTIONAL) |  |

|  |  |
| --- | --- |
| ((MARKS)) (1/2/3...) | 1 |
| ((QUESTION)) | What is the output of this program? |
| ((OPTION\_A)) | Compilation error |
| ((OPTION\_B)) | Run time error |
| ((OPTION\_C)) | Output : a, b and c 10 20 30 |
| ((OPTION\_D)) | None of the mentioned |
| ((CORRECT\_CHOICE)) (A/B/C/D) | A |
| ((EXPLANATION)) (OPTIONAL) |  |

|  |  |
| --- | --- |
| ((MARKS)) (1/2/3...) | 1 |
| ((QUESTION)) | Which of the following supports the concept of hierarchical  classification? |
| ((OPTION\_A)) | Polymorphism (b) Encapsulation  (c) Abstraction (d) Inheritance  Polymorphism (b) Encapsulation  (c) Abstraction (d) Inheritance  Polymorphism |
| ((OPTION\_B)) | Encapsulation |
| ((OPTION\_C)) | Abstraction |
| ((OPTION\_D)) | Inheritance |
| ((CORRECT\_CHOICE)) (A/B/C/D) | D |
| ((EXPLANATION)) (OPTIONAL) |  |

|  |  |
| --- | --- |
| ((MARKS)) (1/2/3...) | 1 |
| ((QUESTION)) | Which language follows the Logical paradigm |
| ((OPTION\_A)) | Prolog |
| ((OPTION\_B)) | Java |
| ((OPTION\_C)) | Fortran |
| ((OPTION\_D)) | cobol |
| ((CORRECT\_CHOICE)) (A/B/C/D) | A |
| ((EXPLANATION)) (OPTIONAL) |  |

|  |  |
| --- | --- |
| ((MARKS)) (1/2/3...) | 1 |
| ((QUESTION)) | Logic programming also known as |
| ((OPTION\_A)) | Imperative programming |
| ((OPTION\_B)) | declarative programming |
| ((OPTION\_C)) | Applicative programming |
| ((OPTION\_D)) | Object oriented programming |
| ((CORRECT\_CHOICE)) (A/B/C/D) | b |
| ((EXPLANATION)) (OPTIONAL) |  |

|  |  |
| --- | --- |
| ((MARKS)) (1/2/3...) | 1 |
| ((QUESTION)) | logic programming does not have code, instead it defines two  pieces of knowledge |
| ((OPTION\_A)) | Head & subhead |
| ((OPTION\_B)) | Code and program |
| ((OPTION\_C)) | facts and rules |
| ((OPTION\_D)) | Loops and subroutines |
| ((CORRECT\_CHOICE)) (A/B/C/D) | C |
| ((EXPLANATION)) (OPTIONAL) |  |

|  |  |
| --- | --- |
| ((MARKS)) (1/2/3...) | 1 |
| ((QUESTION)) | The first procedure programming language was |
| ((OPTION\_A)) | Ada |
| ((OPTION\_B)) | FORTRAN |
| ((OPTION\_C)) | C |
| ((OPTION\_D)) | LISP |
| ((CORRECT\_CHOICE)) (A/B/C/D) | B |
| ((EXPLANATION)) (OPTIONAL) |  |

|  |  |
| --- | --- |
| ((MARKS)) (1/2/3...) | 1 |
| ((QUESTION)) | The first operational functional programming language was |
| ((OPTION\_A)) | Ada |
| ((OPTION\_B)) | FORTRAN |
| ((OPTION\_C)) | ALGOL |
| ((OPTION\_D)) | LISP |
| ((CORRECT\_CHOICE)) (A/B/C/D) | D |
| ((EXPLANATION)) (OPTIONAL) |  |

|  |  |
| --- | --- |
| ((MARKS)) (1/2/3...) | 1 |
| ((QUESTION)) | Iteration in functional languages is usually accomplished via  recursion |
| ((OPTION\_A)) | true |
| ((OPTION\_B)) | false |
| ((OPTION\_C)) |  |
| ((OPTION\_D)) |  |
| ((CORRECT\_CHOICE)) (A/B/C/D) | A |
| ((EXPLANATION)) (OPTIONAL) |  |

|  |  |
| --- | --- |
| ((MARKS)) (1/2/3...) | 1 |
| ((QUESTION)) | In Functional Programming variables are mutable |
| ((OPTION\_A)) | true |
| ((OPTION\_B)) | false |
| ((OPTION\_C)) |  |
| ((OPTION\_D)) |  |
| ((CORRECT\_CHOICE)) (A/B/C/D) | B |
| ((EXPLANATION)) (OPTIONAL) |  |

|  |  |
| --- | --- |
| ((MARKS)) (1/2/3...) | 1 |
| ((QUESTION)) | Functional programming consists only of |
| ((OPTION\_A)) | statements |
| ((OPTION\_B)) | Pure functions |
| ((OPTION\_C)) | functions |
| ((OPTION\_D)) | Lower order functions |
| ((CORRECT\_CHOICE)) (A/B/C/D) | b |
| ((EXPLANATION)) (OPTIONAL) |  |