

**Code : 051610**

**B.Tech. 6th Semester Exam., 2014**

**PRINCIPLES OF PROGRAMMING  
LANGUAGES**

Time : 3 hours

Full Marks : 70

Instructions :

- (i) **All** questions carry equal marks.
- (ii) There are **NINE** questions in this paper.
- (iii) Attempt **FIVE** questions in all
- (iv) Question No. **1** is compulsory.

**1.** Choose the correct answer (any seven) :

- (a) Attribute grammars are used to specify
  - (i) basic syntax of a programming language
  - (ii) non-finite state machines
  - (iii) static semantics of a programming language
  - (iv) dynamic semantics of a programming language

**14AK—400/690**

( Turn Over )

- (b) Assuming that  $y$  is of type String, what happens in the Java statement 'String  $x = y$ '?
- (i) The wrapper method for a String is called to create a new String object and the result bound to  $x$
  - (ii) The address of  $y$  is modified to be the address of  $x$
  - (iii) The object bound to  $y$  is copied and bound to  $x$ , and any previous binding of  $x$  to an object is lost
  - (iv)  $x$  and  $y$  become aliases
  - (v) The contents of  $y$  are copied into the storage allocated to  $x$

- (c) Which of the following situations will create a 'dangling pointer'?
- (i) Setting an arbitrary pointer variable to 'null'
  - (ii) Setting the CDR part of a CONS cell somewhere in the middle of a list to 'null'
  - (iii) Freeing a block to which there still exists a live pointer
  - (iv) Freeing a block which still contains pointers to other existing blocks

14AK-400/690

( Continued )

- (d) What best distinguishes a purely 'functional' programming language from an 'imperative' one?
- (i) There are no variables and hence no assignment operation in a purely functional language
  - (ii) A purely functional language lacks the 'go to' statement, but an imperative language always has such a command
  - (iii) All subprograms must be declared with the keyword function in a purely functional language
  - (iv) There is no real difference, only a difference in the recommended coding style
- (e) The main difference between a sentence and a sentential form is
- (i) there is no difference
  - (ii) a sentence contains only terminal symbols but a sentential form can contain some non-terminal symbols
  - (iii) sentential forms are a subset of sentences but the converse is not true
  - (iv) sentential forms have no handles but a sentence does

14AK-400/690

( Turn Over )

- (f) The language that is generally regarded as the first object-oriented language is
- (i) Java
  - (ii) C++
  - (iii) Simula
  - (iv) LISP
  - (v) COBOL
  - (vi) Ada
  - (vii) Logo
- (g) Axiomatic semantics is often used for
- (i) program verification
  - (ii) byte code generation
  - (iii) syntax-directed parsing
  - (iv) encoding an unambiguous language specification
- (h) Operator associativity generally applies to
- (i) only prefix operators
  - (ii) only infix operators
  - (iii) only postfix operators
  - (iv) prefix, infix and postfix operators

14AK-400/690

( Continued )

- (i) A tail-recursive program is generally better than a nontail-recursive program, because
- (i) it can be run without growing the stack
  - (ii) it is easier to understand
  - (iii) it has no side effects
  - (iv) All of the above
- (j) Dynamic-type checking adds some execution-time overhead, but improves reliability of programs.
- (i) True
  - (ii) False
2. In the following table, the first column lists some s-expressions we would like to construct. Complete the rest of the table by giving the simplest possible expression to build the s-expression. The expression in the second column should only use the CONS function and the one in the third should only use LIST. If it is not possible to construct an s-expression, say so. You are not permitted to include a quoted list (like '(b c)) in giving your answer.

14AK-400/690

( Turn Over )

You are permitted, of course to use quoted atoms (e.g., 'A) and also to use NIL (which need never be quoted) :

s-EXP	Using CONS	Using LIST
(a b)		
((a))		
(a (b c))		
(a (b . c))		
(( ) (( )))		

3. (a) What do the following expressions evaluate?

- (i) ((lambda nil 33))
- (ii) ((lambda (x y)(+x y)) 44 ((lambda (x y)(+x y)) 22 33))
- (iii) (funcall (lambda (x y z) (\* z(+x y))) 3 5 7)

(b) Suppose we defined a function MAPCDR as follows :

```
(defun mapcdr (f1)
  (if (null 1)
      NIL
      (cons (funcall f1) (mapcdr f (cdr 1)))))
```

What would be the following expressions return? If you believe that the expression would cause an error, say so :

- (i) (mapcdr #'length '(a b c d))
- (ii) (mapcdr #'car '(a b c d))
- (iii) (mapcdr #'(lambda (x) x) '(a b c d))

4. Consider the following code fragment in a new language currently under design called ALGORE, which has Pascal-like syntax except that all procedures have no formal parameters. The designers cannot decide whether to adopt dynamic or static scoping :

```
program test;
var a, b: integer;
procedure sub1;
  var a, c, e: integer;
begin
  a := 6; c := 7; e := 8; ... {position 3}
end; {sub1}
procedure sub2;
var a, c, d: integer;
begin
  a := 3; c := 4; d := 5; ... {position 2}
sub1();
...
end; {sub2}
begin
  a := 1; b := 2; ... {position 1}
sub2();
...
end. {test}
```

(a) Determine what variables are visible at positions 1, 2, 3 and what their values are (given the call sequence indicated) with static scoping.

Code : 051610

B.Tech 6th Semester Exam., 2015

PRINCIPLES OF PROGRAMMING  
LANGUAGES

Time : 3 hours

Full Marks : 70

Instructions :

- (i) The marks are indicated in the right-hand margin.  
 (ii) There are **NINE** questions in this paper.  
 (iii) Attempt **FIVE** questions in all.  
 (iv) Question No. 1 is compulsory.

1. Choose the correct option of the following  
 (any seven) :  $2 \times 7 = 14$

(a) The most appropriate matching for the pairs

X : Indirect addressing	1. Loops
Y : Immediate addressing	2. Pointers
Z : Auto-decrement addressing	3. Constants

is

- (i) X-3, Y-2, Z-1  
 (ii) X-1, Y-3, Z-2  
 (iii) X-2, Y-3, Z-1  
 (iv) X-3, Y-1, Z-2

( Turn Over )

AK15-820/590

(b) Aliasing in the context of programming languages refers to

- (i) multiple variables having the same memory location
- (ii) multiple variables having the same value
- (iii) multiple variables having the same identifier
- (iv) multiple uses of the same variable

(c) What is printed by the print statements in the following program P1 assuming call by reference parameter passing?

```

Program P1()
{
  x=10;
  y=3;
  func1(y, x, x);
  print x;
  print y;
}
func1 (x, y, z)
{
  y=y+4;
  z=x+y+z;
}

```

- (i) 10, 3
- (ii) 31, 3
- (iii) 27, 7
- (iv) None of the above

AK15-820/590

(Continued)

(d) Consider the following program

```

ProgramP2
varn: int;
procedureW(varx: int)
begin
  x=x+1;
  print x;
end

procedureD
begin
  varn: 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?

- (i) 10
- (ii) 11
- (iii) 3
- (iv) None of the above

AK15-820/590

(Turn Over)

(e) The results returned by functions under value-result and reference parameter passing conventions

- (i) do not differ
- (ii) differ in the presence of loops
- (iii) differ in all classes
- (iv) may differ in the presence of exceptions

(f) Consider the following program in C language :

```
#include <stdio.h>
main()
{
    inti;
    int*pi = &i;
    scanf("%d", pi);
    printf("%d\n", i+5);
}
```

Which one of the following statements is true?

- (i) Compilation fails
- (ii) Execution results in a run-time error
- (iii) On execution, the value printed is 5 more than the address of variable i
- (iv) On execution, the value printed is 5 more than the integer value entered

K15—820/590

( Continued )

(g) Consider the function func shown below :

```
intfunc(intnum)
{
    intcount = 0;
    while(num)
    {
        count++;
        num>>=1;
    }
    return(count);
}
```

The value returned by func(435) is

- (i) 7
- (ii) 8
- (iii) 9
- (iv) 10

(h) Consider the C function given below :

```
intf(inti)
{
    staticinti = 50;
    intk;
    if(i==j)
    {
        printf("something");
        k = f(i);
        return0;
    }
    elsereturn0;
}
```

Which one of the following is true?

- (i) The function returns 0 for all values of j

AK15—820/590

( Turn Over )

- (ii) The function prints the string something for all values of j  
 (iii) The function returns 0 when j = 50  
 (iv) The function will exhaust the run-time stack or run into an infinite loop when j = 50

(v) Consider the C function given below. Assume that the array listA contains n (> 0) elements sorted in ascending order :

```
intProcessArray(int*listA, intx, intn)
{
    inti, j, k;
    i = 0;
    j = n-1;
    do{
        k = (i+j)/2;
        if(x <= listA[k])
            j = k-1;
        if(listA[k] <= x)
            i = k+1;
    } while(i <= j);
    if(listA[k] == x)
        return(k);
    else
        return -1;
}
```

Which one of the following statements about the function ProcessArray is correct?

- (i) It will run into an infinite loop when x is not in listA

AK15—820/590

( Continued )

- (ii) It is an implementation of binary search  
 (iii) It will always find the maximum element in listA  
 (iv) It will return -1 even when x is present in listA

(v) Consider the following function :

```
doublef(doublex)
{
    if(abs(x*x - 3) < 0.01) returnx;
    elsereturnf(x/2 + 1.5/x);
}
```

Give a value q (to 2 decimals), such that f(q) will return

- (i) q : 1.854  
 (ii) q : 1.732  
 (iii) q : 1.975  
 (iv) q : 1.284

2. (a) What is polymorphism? How can run-time and compile-time polymorphisms be achieved? Explain with program example.

(b) Write a program in C++/Java to create a class Vehicle, the properties of which are—vehicle number, number of wheels and maximum speed. Also create two child classes—Car (new property is—

AK15—820/590

( Turn Over )



number of passengers) and Truck (new property is—load limit). Now create several objects of both the child classes and find out the average speed of which type is greater. 7

3. (a) State different types of inheritances. What are the differences between abstract class and interface? 3+4=7

(b) Write a program to create an abstract class Employee, the abstract method of which is salary(). Now create a child class to inherit the class Employee and define its abstract method. 7

4. (a) Find out the output of the following code : 7

```
#include<iostream.h>
int n=10;
void main()
{
int n=5;
{
int n=20;
cout<<"Inside block \n"<<n;
}
cout<<"Outside block \n"<<n;
cout<<"Outside block and scope
resolved \n"<<n;
}
```

(b) Explain different storage classes in C language. 7

AK15—820/590

( Continued )

5. (a) What is heap? Create a heap with the following numbers : 7

24, 45, 76, 49, 65, 29, 50

Now sort the heap.

(b) Explain the organization of a conventional computer with proper block diagram. 7

6. Write short notes on the following : 7+7=14

(a) Firmware computer

(b) ADT

7. (a) Explain call by value and call by reference with example. 7

(b) Describe scalar and composite data types. 7

8. (a) Describe BNF notation. 7

(b) What are the different types of grammar for programming languages? 7

9. (a) Illustrate well-known skeleton patterns for parallel programming with example. 7

(b) What are the different translation phases for programming languages? Write the characteristics of translation phases. 3+4=7

\*\*\*

AK15—820/590

Code : 051610

9. Write short notes on the following : 7+7=14

- (a) BNF
- (b) Exception handling in C++

\*\*\*

akubihar.com

**B.Tech 6th Semester Exam., 2016**

**PRINCIPLES OF PROGRAMMING  
LANGUAGES**

Time : 3 hours

Full Marks : 70

Instructions :

- (i) The marks are indicated in the right-hand margin.
- (ii) There are **NINE** questions in this paper.
- (iii) Attempt **FIVE** questions in all.
- (iv) Question No. 1 is compulsory.

akubihar.com

akubihar.com

akubihar.com

1. Choose the correct choice for each of the following (any seven) : 2×7=14
  - (a) What is printed by the print statements in the program P1 assuming call by reference parameter passing?

```

Program P1{
    x=10; y=3; func 1(y, x, x);
    print x; print y;}

func 1(x, y, z){
    y=y+4; z=x+y+z;}

```

- (i) 10, 3
- (ii) 31, 3
- (iii) 27, 7
- (iv) None of the above

akubihar.com

akubihar.com

akubihar.com

akubihar.com

akubihar.com

(b) Which of the following is the functionality of 'data abstraction'?

- (i) Reduce complexity
- (ii) Binds together code and data
- (iii) Parallelism
- (iv) None of the above

(c) Which languages necessarily need heap allocation in the run-time environment?

- (i) Those that support recursion
- (ii) Those that use dynamic scoping
- (iii) Those that allow dynamic data structures
- (iv) Those that use global variables

(d) Which of the following statements are correct?

1. Static allocation of all data areas by a compiler makes it impossible to implement recursion.
2. Automatic garbage collection is essential to implement recursion.
3. Dynamic allocation of activation records is essential to implement recursion.

4. Both heap and stack are essential to implement recursion.

- (i) 1 and 2 only
- (ii) 2 and 3 only
- (iii) 3 and 4 only
- (iv) 1 and 3 only

(e) The most appropriate matching for the following pairs is

X : m=malloc(5); m=NULL; 1 : using dangling pointers  
 : free(n); n->value=5; 2 : using uninitialized pointers  
 : char \*p; \*p='a'; 3 : lost memory

- (i) X-1, Y-3, Z-2
- (ii) X-2, Y-1, Z-3
- (iii) X-3, Y-2, Z-1
- (iv) X-3, Y-1, Z-2

(f) The following program fragment is written in a programming language that allows variables and does not allow nested declarations of functions :

```
global int i=100, j=5;
void P(x){
    int i=10;
    print(x+10);
    i=200; j=20;
    print(x);
}
main(){
    P(i+j);
}
```

If the programming language uses static scoping and call by need parameter passing mechanism, the values printed by the above program are

- (i) 115, 220
- (ii) 25, 220
- (iii) 25, 15
- (iv) 115, 105

(g) The attributes of three arithmetic operators in some programming language are given below :

Operator	Precedence	Associativity	Arity
+	High	Left	Binary
-	Medium	Right	Binary
*	Low	Left	Binary

The value of the expression  $2 - 5 + 1 - 7 * 3$  in this language is

- (i) 1
- (ii) 2
- (iii) 3
- (iv) 9

(h) Operator associativity generally applies to

- (i) only prefix operators
- (ii) only infix operators
- (iii) only postfix operators
- (iv) prefix, infix and postfix operators

(i) In JavaScript, the expression  $A ? B : C$  is the equivalent to

- (i) if (A) {B} else {C}
- (ii) if (A==B) C;
- (iii) if (A!=B) C;
- (iv) if (A) {B; C};

(j) In an object-oriented language, when an object has several methods have the same name but different signatures, we call this

- (i) inheritance
- (ii) overriding
- (iii) overloading
- (iv) signature clash

2. (a) Discuss the formal methods of describing syntax. 7

(b) Explain programming paradigms. 7

3. (a) Write a short note on referencing environment with example. 7

(b) Briefly explain various methods of passing parameter to subprograms with suitable examples. 7

4. (a) Explain the functional programming language and its applications. 7

(b) Is the following grammar ambiguous? If not, give your argument. If yes, give an example that illustrates the ambiguity, and then modify the grammar to remove the ambiguity: 7

$\text{Expr} \rightarrow \text{Expr} + \text{Expr} \mid \text{Expr} - \text{Expr} \mid$

$(\text{Expr}) \mid \text{Integer}$

$\text{Integer} \rightarrow 0 \mid \dots \mid 9$

5. (a) Consider the following program :

```

1. int i=0
2. void A() {
3.     print i;
4. }
5. void B() {
6.     int i;
7.     i=2;
8.     A();
9. }
10. void main(int a) {
11.     if (a>0) B();
12.     else A();
13. }
```

Answer the following questions in both static and dynamic scoping :

(i) What does it outputs in an execution main(3)?

(ii) What is the output of an execution main(0)? 7

(b) Explain various design issues associated with object-oriented languages. 7

6. (a) Briefly explain the themes of imperative programming, functional programming and logic programming. 7

(b) What are different types of error that can occur in a program? Give one example of each. 7

7. (a) In general, it is not possible to check statically that the r-value of an uninitialized variable will not be used during program execution. Why? 7

(b) What is the difference between macros and routines? Explain the difference in terms of the concept of binding. 7

8. (a) Differentiate between system defined data types and abstract data types with suitable examples. 7

(b) Discuss two main approaches to identify and reuse free-memory area in a heap. 7

## B.Tech 6th Semester Examination, 2017

## Principles of Programming Languages

Full Marks : 70

Time : 3 hours

## Instructions :

- (i) There are *Nine Questions in this Paper.*
- (ii) *Attempt Five questions in all.*
- (iii) **Question No. 1 is Compulsory.**
- (iv) *The marks are indicated in the right-hand margin.*

1. Choose the correct choice for each of the following (any seven):

2×7=14

akubihar.com

(a) The results returned by functions under value-result and reference parameter passing conventions.

- (i) Do not differ
- (ii) Differ in the presence of loops
- (iii) Differ in all cases
- (iv) May differ in the presence of exceptions

(b) Consider the C code to swap two integers and these five

statements:

```
void swap(int *px, int *py) {
```

```
    *px = *px - *py;      akubihar.com
```

```
    *py = *px + *py;
```

	Group -1	Group-2
P	Functional	1. Command-based, procedural
Q	Logic	2. Imperative, abstract data type
R	Object-oriented	3. Side-effect free, declarative, expression evaluation
S	Imperative	4. Declarative, clausal representation, theorem proving

- (i) P-2, Q-3, R-4, S-1  
 (ii) P-4, Q-3, R-2, S-1  
 (iii) P-3, Q-4, R-1, S-2  
 (iv) P-3, Q-4, R-2, S-1

akubihar.com

(e) Scheme macros are primarily used to define:

- (i) Dynamically scoped environments  
 (ii) Closures  
 (iii) Functions that don't evaluate all of their arguments  
 (iv) Reflective programs

(f) In Scheme, a tail-recursive algorithm is generally better than a non-tail recursive algorithm because

- (i) It can be run without growing the stack  
 (ii) It is easier to understand  
 (iii) It has no side effects  
 (iv) All of the above.

akubihar.com

Code : 051610

3

P.T.O.

\*px = \*py - \*px;

akubihar.com

S1: will generate a compilation error

S2: may generate a segmentation fault at runtime depending on the arguments passed

S3: correctly implements the swap procedure for all input pointers referring to

integers stored in memory locations accessible to the process

akubihar.com

S4: implements the swap procedure correctly for some but not all valid input pointers

S5: may add or subtract integers and pointers

- (i) S1  
 (ii) S2 and S3  
 (iii) S2 and S4  
 (iv) S2 and S5

akubihar.com

(c) Which one of the following is NOT performed during compilation ?

- (i) Dynamic memory allocation  
 (ii) Type checking  
 (iii) Symbol table management  
 (iv) Inline expansion

akubihar.com

(d) Choose the best matching between the programming styles in Group 1 and their characteristics in Group 2

Code : 051610

2

(g) Most modern programming languages use the following schemes for their variable scoping

- (i) Lexical scoping
- (ii) Dynamic scoping
- (iii) Both lexical and dynamic scoping
- (iv) Scope inferencing

akubihar.com

(h) Attribute grammars are used to specify

- (i) Basic syntax of a programming language
- (ii) Non-finite state machines
- (iii) Static semantics of a programming language
- (iv) Dynamic semantics of a programming language

(i) What will be the output of the following pseudo-code when parameters are passed by reference and dynamic

scope is assumed

akubihar.com

```
a = 3;
void n(x) {x = x* a; print (x); }
void m(y) {a = 1; a = y-a; print(a);}
void main () {m(a);}
```

(i) 6.2

(ii) 6.6

(iii) 4.2

(iv) 4.4

akubihar.com

(i) A common property of logic programming languages and functional languages is:

akubihar.com

- (i) both are procedural languages
- (ii) both are based on  $\lambda$ -calculus
- (iii) both are declarative
- (iv) both use Horn-clauses

2. (a) For the following program fragment, describe each stage in the life of the run-time stack until routine beta is called (recursively), by alpha. In particular, show the dynamic and static links before each routine call.

```
inti = i, j -2, k-3;
akubihar.com
```

beta ();

```
alpha () {int i = 4, l = 5;
```

```
    i += k + 1;
    beta();
```

```
    ...}
```

```
beta() { int k = 6;
```

```
    i = j + k; alpha ();
```

```
    ...}
```

```
main() {
```

```
    ...
```

```
    beta();
```

```
    ...}
```

akubihar.com

7



7. (a) Consider the context-free grammar  $G = (V, T, P, S)$ , where  $V = \{S\}$ ,  $T = \{0, 1\}$ , and the productions  $P$  are defined by:

akubihar.com       $S \rightarrow 0S11S0^*$

Argue that every string generated by this grammar is balanced, i.e., if  $w$  is derived from  $S$ , then  $n_0(w) = n_1(w)$  where  $n_0(w)$  and  $n_1(w)$  stand for the number of 0's and 1's respectively in  $w$ .

(b) Define monomorphic and polymorphic type systems.

8. (a) Explain the features of java.      7  
 (b) Describe the programming language, PROLOG and also write the deficiencies of PROLOG.      7

9. Write Short notes on the following:      7-7  
 (a) Static Allocation      akubihar.com  
 (b) Client Server Computing      \*\*\*

(b) Study the case statement of Pascal and compare it to the C++ switch statement and the Ada case statement.      7

3. (a) What is ambiguous grammar? Explain with example.      7  
 (b) Describe the scope and lifetime of a variable.      7

4. (a) Compare and contrast static binding with dynamic binding.      7  
 akubihar.com

(b) What is coroutine? Explain Briefly.      7

5. (a) What is the difference between overloaded functions and generic functions in C++?      7  
 (b) Sometimes inheritance is used improperly. For example, consider defining an automobile class. We have an existing class window. Since automobiles have windows, we have two options: we can derive automobile from window, ensuring that the automobile will have a window, or we can define the class automobile and use window as a member of the class. Why is the second solution better? Explain the proper use of inheritance in terms of the is-a relation.      akubihar.com      7

6. (a) Java supports both interfaces and abstract classes. Both may be used to define specification. Compare these two concepts as supported in Java.      7  
 (b) Describe stack object allocation and heap object allocation.      7

akubihar.com