Universidade Federal de Roraima

Departamento de Ciência da Computação

Linguagens de Programação

DISCIPLINA: Linguagens de Programação – DCC206

1ª Lista

ALUNO (A): **VANCARLA SOARES AQUINO** NOTA:\_\_\_\_

ATENÇÃO: Descrever as soluções com o máximo de detalhes possível, no caso de programas, inclusive a forma como os testes foram feitos. Todos os artefatos (relatório com as respostas, código fonte de programas, e outros) gerados para este trabalho devem ser adicionados em um repositório no site github.com. Para as questões que requisitarem a escrita/implementação de programas deve ser apresentado: o modo de compilar/executar o programa; a linha de comando para executar o programa; e um exemplo de entrada/saída do programa.

**1) Conceitue e descreva as diferenças entre (i) linguagem de alto-nível e (ii) linguagem de baixo-nível. Descreva o papel e a importância do compilador no processo de criação de programas de computador.**

Linguagem de programação de *alto nível* é uma linguagem com um nível de abstração relativamente elevado, longe do código de máquina e mais próximo à linguagem humana. Ex: JavaScript, Python, PHP.

Linguagem de programação de *baixo nível* é aquela que se aproxima mais da linguagem de máquina, que você precisa ter o conhecimento direto da arquitetura do computador para fazer alguma coisa. Ex: Assembly.

Um compilador traduz um programa de uma linguagem textual facilmente entendida por um ser humano para uma linguagem de máquina, específica para um processador e sistema operacional.

**5) Descreva as seguintes categorias de linguagens de programação e apresente o nome de duas linguagens de programação com seus respectivos exemplos.**

**(A)Imperativas**

Variáveis, atribuição e iteração, Inclui programação orientada a objeto, scripting e visuais. Exemplos: C, Java, Perl, JavaScript, Visual BASIC, C++.

**(B)Funcionais**

Aplicação de funções sob parâmetros. Exemplos: LISP, Scheme, Haskell, Ocaml.

**(C)Lógicas**

Baseada em fatos e regras. Exemplo: Prolog

**(D)Marcação/Híbrida**

Linguagens de marcação estendida para suportar alguma programação. Exemplos: JSTL, XSLT.

**7) Descreva o que é um paradigma de programação.**

Modelo, padrão ou estilo de programação suportado por linguagens que agrupam certas características comuns. A classificação de linguagens em paradigmas é uma consequência de decisões de projeto que tem impacto na forma segundo a qual uma aplicação real é modelada do ponto de vista computacional.

**9) Defina análise semântica e descreva: Semântica Operacional; Semântica Axiomática; e Semântica Denotacional.**

**Análise semântica** é a terceira fase da compilação onde se verificam os erros semânticos, no código fonte e coletam-se as informações necessárias para a próxima fase da compilação, que é a geração de código objeto.

**Semântica operacional** preocupa-se mais em como os programas são executados do que meramente com os resultados destas computações.

**Semântica Axiomática** presta-se particularmente à prova e raciocínio sobre propriedades dos programas, e à sua verificação, i.e. à prova de correção dos programas face às suas especificações.

**Semântica Denotacional** é um método formal, pois é baseado em fundamentos matemáticos bem entendidos e usa uma notação rigorosamente definida (meta-linguagem).

**10) Utilizando a semântica axiomática apresente a pré-condição para os seguintes programas.**

PROGRAMA – A

x = 122 \* y - 144

{x > 144}

Se x > 144 → 122 \* y – 144 > 144

Logo:

122 \* y > 144+144

122y > 288

y > 288/122

{y > 2,3}

PROGRAMA – B

y = 5 \* x - 5

x = y + 5

{x < 45}

Se x < 45 → y + 5 < 45

Então

y + 5 < 45

y < 45 – 5

{y < 40}

Logo

y = 5\* x – 5

5 \*x – 5 < 40

5x < 40+5

5x < 45

x < 45/5

{x < 9}

PROGRAMA – C

if(x < 200)

y = y + 2;

else

y = y – 2;

{y > 2}

Se y > 2 → y + 2 > 2

Então

y + 2 > 2

y > 2 – 2

{y > 0}

Logo

y – 2 > 2

y > 2+2

{y > 4}

PROGRAMA – D

while i < N do

i = i + 1

end

{i == N}

Se i == N → i + 1 == N

{ i < N }

PROGRAMA – E

i = 1; sn = 0; n = 32767;

while (i <= n){

sn = sn + a;

i = i + 1;

}

{sn == n \* a}