مقدمة عن البرمجة والمنطق البرمجي تم التركيز على أهمية تنظيم الكود وفهم مكونات السال بشكل دقيق قبل التنفيذ، مع كر أن المكتبات والبية البرمجية متوفرة ويمكن الاعتماد عليها 1. العمليات الحسابية الأساسية في البرمجة والقسمة، كاركتر) لكل متغير 3. يتم استخدام المتغيرات لتخزين الأرقام والنتاج، مع رورة تحديد نوع البيانات بشكل صحيح، مثل استخدام دبل للأرقام العشرية 3. مثال على جمع رقمين: يتم وضع النتيجة في متغير ريزلت، ويتم باعتها باستخدام أوامر الطباعة مع تنسيق مناسب 4. استخدام الشروط والتحكم في التدفق الشروط تستخدم للتحكم في سير البرنامج، يتم استخدام شرط if مع نفي (علامة التعجب) للتحقق من أن المقام ليس صفر قبل إجراء عملية القسمة 6. في حالة أن المقام يساوي صفر، يتم إظهار رسالة خطأ وتعيين النتيجة إلى صفر لمنع الخطأ الحسابي 7. التحقق من صحة العمليات الحسابية وإلا يتم إظهار رسالة خطأ وتعيين النتيجة إلى صفر 7. يتم استخدام شرط if للتحقق من أن النمبر تو (المقام) لا يساوي صفر، switch يستخدم لتنظيم العمليات الحسابية بناء على نوع العملية (جمع، رح، رب، ويُستخدم في switch مع حالات لكل عملية، الحالات تشمل: مع باعة النتيجة بشكل منسق 4. رح (ـ): يتم رح الرقم الثاني من الأول، مع التحقق من أن الرقم الأول أكبر لضمان النتيجة الصحيحة 11. رب (*): يتم رب الرقمين وتخزين النتيجة 12. قسمة (/): يتم التحقق من أن المقام لا يساوي صفر قبل القسمة، وإذا كان صفر يتم إظهار رسالة خطأ وتعيين النتيجة إلى صفر 13. يتم التحقق من أن الأوبريتور صحيح، وأنه إما +، أو /، مع إظهار رسالة خطأ إذا كان ير لك 15. يتم التحقق من أن المقام لا يساوي صفر، وإذا كان صفر، يتم إظهار رسالة خطأ وتعيين النتيجة إلى صفر 16. ملاحظات مهمة يجب وضع break بعد كل حالة في switch لمنع الانتقال ير المرغوب 17. مثل: "القسمة على صفر ير مسموحة" 7. استخدام التعليقات والتنسيق الجيد يسهل فهم الكود وتصحيح الأخطاء 18. مقدمة عن البرمجة والكود النيم سبيس، حيث أن برامج ات كفاءة عالية تتطلب فهم عميق للهياكل البيانات والخوارزميات 20. المدخلات والمخرجات وcout لعرض النتاج على الشاشة 21. م تحديد إذا كان موجب، أو صفر باستخدام شروط if أو 22 switch. التحقق من القيم المدخلة يتم التحقق من صحة المدخلات، خاصة المقام في العمليات الحسابية، باستخدام if وليس switch، وتجنب الأخطاء في البرنامج 24. التحكم في تدفق البرنامج يمكن استخدام switch لتصنيف العمليات أو القيم المدخلة، وهو مفيد عند وجود حالات متعددة، switch يتطلب أن تكون القيم المدخلة من نوع صحيح (int أو char) ويستخدم break للخروج من الحالة بعد التنفيذ 25. العمليات الحسابية والتعامل مع الأرقام الضرب *، والقسمة /، أو صفر باستخدام if، و28 (n == 0) if. يمكن تتبع البرنامج باستخدام تريسين (التتبع خطوة بخطوة) لفهم تدفق التنفيذ، خاصة عند التعامل مع شروط متعددة 29. المتغيرات وأسماءها مثل عدم وجود مسافات، استخدام underscore _ ، وتجنب الرموز الخاصة مثل # أو @ 30. ويجب أن يبدأ بحرف وليس برمز أو رقم 31. n0، أو استخدام switch عندما تكون الحالات تعتمد على قيمة واحدة محددة 33. العمليات الحسابية المتقدمة مثل 1 =+ n لزيادة n بمقدار واحد 34. ملخص عن البرمجة الجيدة تحسين الأداء يتطلب فهم عميق للهياكل البيانات والخوارزميات، وليس فقط كتابة الكود 36. والتحقق من المدخلات، وتسمية المتغيرات بشكل مناسب، كلها من أساسيات البرمجة الجيدة 38. مفاهيم البرمجة الأساسية: الطباعة باستخدام السي اوت: يتم استخدام سي اوت لطباعة القيم، مع اختلاف الترتيب في التنفيذ (قبل أو بعد الطباعة) 40. مع تأثير الترتيب على القيمة المقروة 41. مع تحديث القيمة في الذاكرة 42. الطرح، والزيادة أو النقصان على المتغيرات، الأولوية في التنفيذ: ترتيب العمليات مهم، خاصة عند استخدام سي اوت مع عمليات حسابية أو عمليات يادة/نقصان، الترتيب في الطباعة: الطباعة تعتمد على ترتيب الأوامر، ستظهر القيمة القديمة، وإذا بعد التعديل، ستظهر القيمة الجديدة 45. العمليات الشرطية والتكرار: مثل التحقق من قيمة متغير معين 41. التكرار: استخدام الحلقات مثل for أو while لزيادة أو نقصان قيمة المتغيرات بشكل متكرر، مع ملاحظة أن ++ و ـــ تثر على القيم بشكل مباشر 40. ملاحظات مهمة: التركيز على الترتيب: الترتيب في كتابة الأوامر مهم جدا،

lakhasly.com © تم تلخيص النص بواسطة موقع لخصلي