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 if الشروط والتحكم في التدفق الشروط تستخدم للتحكم في سير البرنامج، يتم استخدام شرط 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 switch ،للتحقق من أن النمبر تو (المقام) لا يساوي صفر if شرط يستخدم لتنظيم العمليات الحسابية بناء على نوع العملية (جمع، مع حالات لكل عملية، الحالات تشمل: مع باعة النتيجة بشكل منسق 4. رح (ـ): يتم رح الرقم switch رح، رب، ويُستخدم في 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 لمنع الانتقال ير المرغوب 17. مثل: "القسمة على صفر ير مسموحة" 7. استخدام التعليقات والتنسيق الجيد switch حالة في يسهل فهم الكود وتصحيح الأخطاء 18. مقدمة عن البرمجة والكود النيم سبيس، حيث أن برامج ات كفاءة عالية تتطلب فهم عميق لعرض النتاج على الشاشة 21. م تحديد إذا كان موجب، أو coutللهياكل البيانات والخوارزميات 20. المدخلات والمخرجات و التحقق من القيم المدخلة يتم التحقق من صحة المدخلات، خاصة المقام في العمليات .22 switch أو if صفر باستخدام شروط switch وتجنب الأخطاء في البرنامج 24. التحكم في تدفق البرنامج يمكن استخدام ،switch وليس if الحسابية، باستخدام switch ،لتصنيف العمليات أو القيم المدخلة، وهو مفيد عند وجود حالات متعددة يتطلب أن تكون القيم المدخلة من نوع صحيح للخروج من الحالة بعد التنفيذ 25. العمليات الحسابية والتعامل مع الأرقام الضرب *، والقسمة /، break ويستخدم (char أو int) يمكن تتبع البرنامج باستخدام تريسين (التتبع خطوة بخطوة) لفهم تدفق التنفيذ، خاصة .28 (n == 0) if،و if أو صفر باستخدام underscore _ ، وتجنب الرموز عند التعامل مع شروط متعددة 29. المتغيرات وأسماءها مثل عدم وجود مسافات، استخدام switch أو استخدام ،n0 .الخاصة مثل # أو @ 30. ويجب أن يبدأ بحرف وليس برمز أو رقم 31 بمقدار واحد 34. ملخص عن البرمجة الجيدة تحسين n لزيادة 1 =+ n قيمة واحدة محددة 33. العمليات الحسابية المتقدمة مثل 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. لزيادة أو نقصان قيمة المتغيرات بشكل متكرر، مع ملاحظة أن ++ و ـــ تثر على while أو for التكرار: استخدام الحلقات مثل ،القيم بشكل مباشر 40. ملاحظات مهمة: التركيز على الترتيب: الترتيب في كتابة الأوامر مهم جدا