تأثير تدريس حل المشكلات غير الرياضية على الفاعلية الذاتية والأداء في مقرر مقدمة إلى البرمجة

د. رياض عبد الرحمن الحسن
جامعة الملك سعود– المملكة العربية السعودية
alhassan@ksu.edu.sa

ملخص: هدفت هذه الدراسة إلى تقصي أثر تدريس مهارات حل المشكلات غير الرياضية على الفاعلية الذاتية والأداء الأكاديمي للطلاب في مقرر مقدمة إلى البرمجة. وقد اقتصرت هذه الدراسة على عينة من الطلاب الذكور الدارسين في مقرر برمجة الحاسبات (1)، والمسجلين خلال الفصل الدراسي الثاني من العام الجامعي2013. وقد اتبعت هذه الدراسة المنهج شبه التجريبي المتضمن اختباراً قبلياً وبعدياً، ومجموعتين؛ احداهما تجريبية والأخرى ضابطة. وقد أجريت الدراسة على شعبتين من مقرر مقدمة إلى البرمجة، وقد أتم الطلاب في كلا المجموعتين مقياس الفاعلية الذاتية في برمجة الحاسبات قبل الشروع في تدريس المقرر. وقد تم تدريس المجموعة الضابطة باستخدام أسلوب التدريس الذي يتبعه أستاذ المقرر عادة في تدريسه، دون تقديم تدريب خاص حول حل المشكلات. أما المجموعة التجريبية فقد تلقت تدريساً إضافياً في حل المشكلات في بداية الفصل الدراسي وتم التركيز على مهارات التفكير الناقد اللازمة لحل المسائل البرمجية. وفي منتصف الفصل الدراسي، ونهايته تم تقديم مقياس الفاعلية الذاتية في برمجة الحاسبات مرة أخرى. وقد أظهرت نتائج الدراسة: حصول المجموعة التجريبية على نقاط أعلى من المجموعة الضابطة على مقياس الفاعلية الذاتية في برمجة الحاسبات، وكذلك حصول المجموعة التجريبية على درجات أعلى في مجال التحصيل لأكاديمي، وقد كانت تلك الفروق بين المجموعتين دالة احصائياً.

الكلمات المفتاحية: الفاعلية الذاتية في برمجة الحاسبات، حل المشكلات غير الرياضية، برمجة الحاسب

مقدمة

يعتبر تدريس المهارات الأساسية في برمجة الحاسبات مشكلة في التعليم منذ مدة طويلة (Kranch, 2010)، فوفقاً ل بندسين وكاسبيرسن (Bennedesen & Caspersen, 2008) فإن تعليم برمجة الحاسبات للمتعلمين المبتدئين يعتبر مشكلة منذ ما يربو على الأربعين عاماً. أما روبينز، ورونتري، ورونتري (Robins, Rountree, & Rountree, 2003) فقد لاحظوا أن الطلاب بشكل عام يعتبرون مقررات البرمجة صعبة، وأن هذه المقررات تتصف بقدر عال من الانسحاب من المقرر، وأن السؤال حول مالذي يمكن عمله

لتخريج مبرمجين افضل قد شغل بال المختصين في مجال البرمجة منذ السبعينات الميلادية. ووفقاً لمكقيترك، و بويل، وابيت، ولويد، ولوفقروف، وماندير (McGettrick, Boyle, Ibbett, Lloyd, Lovegrove, & Mander, 2005) فإن المربين يربطون بين ضعف الاحتفاظ بالمهارات لدى طلاب برامج علوم الحاسب و الإخفاق في مقررات البرمجة الأساسية، والكره العام للبرمجة الذي تولده تلك البرامج. وقد اعتبر مكقتريك وزملاؤه (McGetrick et al., 2005) أن ذلك يعتبر مشكلة كبيرة، لدرجة أنهم قد صنفوا التدريس الفعال لمهارات البرمجة كأحد التحديات السبع الكبرى في عالم الحاسب الآلي. وقد عزى جرايس (Gries, 1974 كما في Kranch, 2010) الصعوبة في تدريس البرمجة إلى إهمال تدريس مهارات حل المشكلات العامة.

وينظر إلى أسلوب حل المشكلات باعتباره استراتيجية تعليمية تهدف إلى ربط المشكلات الواقعية بعملية التعلم، وتتم ضمن مجموعات طلابية تناقش المشكلات وتحاول: الوصول إلى الحل المناسب لها من خلال تطبيق القوانين والعلاقات المناسبة، وممارسة أنشطة تعليمية مختلفة من جمع بيانات يمكن عن طريقها توضيح المشكلة، وتحديد المطلوب ايجاده، والوصول إلى النتائج وتفسيرها، وتكون مهمة المعلم هي تسهيل عملية التعلم وتشجيعها، ومستشاراً لمجموعات النقاش الطلابية، مما يوفر تفاعلاً بين الطلبة والمعلومات، وليس انتقالاً للمعلومات من قبل المعلم إلى الطلبة (نجم، 2011).

وعندما أعطى كل من سولواي، وايرليتش، وبونار، وقرينسبان (Soloway, Ehrlich, Bonar, & Greenspan, 1982) لمجموعة من الطلاب قد أمضوا فصلاً دراسياً كاملاً في دراسة لغة الباسكال ما وصفوه بأنه مثال بسيط على على مشكلة برمجية تتضمن استخدام الحلقات (Loops) لأجل حساب المتوسط الحسابي، وجدو أن 44% من الطلاب لم يتمكنوا من كتابة البرنامج شكل صحيح. وقد لاحظ وينسلو (Winslow, 1996) أن العديد من الدراسات قد أظهرت أن الطلاب الذين أكملوا مقرر المقدمة في البرمجة لم يستوعبوا مفهوم الحلقات البسيطة. وحتى بعد سنتين من دراسة البرمجة، فإن الطلاب في دراسة كرلاند، وبيا، وكلمنت، وماوباي (Kurland, Pea, Clement, & Mawby, 1989) قد حققوا قدراً متواضعاً من فهم لغة البرمجة، ولكنهم يظلون على مستوى منخفض من فهم المفاهيم البرمجية. أما لين ودالبي (Linn & Dalbey, 1989) فقد أجريا دراسة على طلاب المرحلة المتوسطة الذين يدرسون إحدى لغات البرمجة في عدد من المدارس، ووجدا أن معظم الطلاب قد حققوا تقدماً محدوداً جداً في البرمجة، وقليل من الطلاب فقط قد تخطى الفهم البسيط لخصائص لغة البرمجة. ووجد ايجكسترا (Dijkstra, 1969) بأن إحدى الصعوبات في تعلم البرمجة تظهر في الفرق الشاسع بين الأمثلة البسيطة التي يؤديها الطالب في غرفة الصف،

والبرامج الهائلة التي نحتاجها لحل مشكلة حقيقية. أما بوتنام، وسليمان، وباكستر، وكوسبا (Putnam, Sleeman, Baxter, & Kuspa, 1983) فيرون بأن الصعوبات التي يواجهها طلاب المرحلة الثانوية في تتبع البرامج وتوقع المخرجات منها قد جعلتهم يتساءلون عن مستوى الفهم الذي يقدمه لهم المقرر الدراسي.

وتعد تنمية مهارات القدرة على حل المشكلات لدى طلاب تخصص الحوسبة إحدى أهم الأهداف التي تسعى إليها كليات علوم الحاسب، ومطلباً أساسياً لحدوث التنمية البشرية، وتظهر هذه الأهمية في ربط المشكلات التدريسية بالحياة العملية وجعلها مشابهة لمشكلات الحياة اليومية، وذات علاقة باهتمام الطلاب وميولهم، وإتاحة الفرصة الكافية لهم للتفكير بحرية، والتجريب العلمي، والتحديد الوصفي للمشكلة، والتخطيط الهادف لحلها من خلال عرض الفروض، وتفسير البيانات، والوصول إلى تنمية يتم تطبيقها في مواقف جديدة (شريف، 2011).

مشكلة الدراسة

اتضح للباحث من خلال الملاحظة الشخصية، ومراجعة الدراسات السابقة (Heart et al., 2008; Hazzan, Gal-Ezer, & Blum, 2008) أن مادة برمجة الحاسبات ما زالت تقدم للطلاب في كلية الحاسب، والمدارس الثانوية بالطريقة التقليدية والتي تركز على اكتساب مهارات البرمجة دون توافر المعنى والفهم الكافي لها، وعدم توفير فرص حقيقية للطلاب ليمارسوا المهارات البرمجية والعقلية المتضمنة بها، مما أدى إلى إعاقة وتدني مستوى أداء الطلاب لتلك المهارات الهامة، وكذلك قدرتهم على حل المشكلات البرمجية، وعدم تقبلهم العمل التعاوني مع بعضهم البعض.

كما تكمن مشكلة الدراسة فيما لاحظه الباحث من مستوى الانسحاب والرسوب العالي في مقررات المقدمة إلى البرمجة التي تقدم للطلاب في السنة الأولى في تخصص علوم الحاسب، وما لاحظه كذلك من خلال دراسة استطلاعية غير منشورة من مدى الصعوبة التي يواجهها الطلاب في المرحلة الثانوية عند تعلمهم للغة برمجة الحاسب فيجول بيسك. فالانسحاب من مقررات الحاسب الأولية يؤدي عادة إلى تغيير تخصص الطالب من علوم الحاسب إلى تخصص آخر. وهذا بدوره يؤدي إلى قلة المتخصصين في علوم الحاسب بالرغم من النمو في سوق العمل لهذا القطاع. وقد لاحظ بكويث وزملاؤه (Beckwith et al., 2006) أن الانطباع الأول أو التجربة الأولى مع برمجة الحاسبات تؤسس الفاعلية الذاتية للطلاب في مجال البرمجة، وأن الإخفاق المبكر أو مجرد الشعور بالإخفاق يؤسس لفاعلية ذاتية منخفضة في المستقبل على الرغم من احتمال النجاح.

ومن ناحية أخرى، تمثل طرائق وأساليب التدريس أحد العناصر الأساسية في العملي التعليمية كونها تعمل على إيصال الخبرات والأنشطة التعليمية والمهارات للمتعلمين بهدف مساعدتهم

على النمو المعرفي الصحيح في ضوء فلسفة المجتمع وأهدافه التربوية المحددة، كذلك، فإن هذه الطرائق أو أساليب التدريس تمثل الوسيلة الناقلة للعلوم والمعرفة والمهارة للمتعلم، فكلما كانت ملائمة للموقف التعليمي ومنسجمة مع العمر الزمني للمتعلم وقدراته، وميوله كانت الأهداف المتحققة عبرها أوسع عمقاً وأكثر فائدة. ومن بين هده الطرائق والأساليب طريقة حل المشكلات التي تعمل على تنمية التفكير العلمي للمتعلم وايصاله إلى حالة الإبداع والابتكار. إن هذه الطريقة يمكن أن تحقق أهداف العملية التعليمية من خلال مساعدة المتعلم على اكتساب المفاهيم والأسس العامة واتقانها وتنمية قدراته على حل المشكلات التي يواجهها في التعلم أو في حياته العامة ويحقق ارتفاعاً في مستوى اعتزازه بذاته، وينمي عامل الابتكار لديه من خلال تقديم أفكار جديدة وإعادة ترتيب الأفكار السابقة (حسن، 2012).

ولكن من الملاحظ أن العدد المحدود من الدراسات حول حل المشكلات قد ركز على الرياضيات كوسيلة لقياس تلك المهارات، مما يجعل التدريب على حل المشكلات بإسلوبه الحالي منحازاً إلى الرياضيات ويعطي مؤشراً غير دقيق عن القدرات البرمجية للطالب. ويؤكد ليمير (Lemire, 2002) ونجم (2011) بأن هناك شحاً في الدراسات التي تؤكد الفكرة بأن مهارات حل المشكلات التي يتعلمها الطالب في أحد المجالات، كالرياضيات مثلاً، يمكن تطبيقها في مجالات أخرى دون الحاجة إلى تدريب إضافي. وقد ضرب ليمير العديد من الأمثلة التي تظهر عدم قابلية مهارات حل المشكلات للتطبيق في مجالات أخرى، ويقترح أنه لأجل أن تكون مهارات حل المشكلات قابلة في شتى المجالات، فإنه يجب تدريس تلك المهارات ضمن ذلك المجال. ويرى علي (Ali, 2005) بإن مهارات التفكير الناقد و حل المشكلات مهمة جداً في مجال علوم الحاسب، وأنه يجب أت تصبح جزءاً لا يتجزأ من المنهج الدراسي لعلوم الحاسب.

ويمكن تلخيص مشكلة الدراسة في السؤال التالي: ما أثر تدريس مهارات حل المشكلات غير الرياضية على الفاعلية الذاتية والأداء الأكاديمي للطلاب في مقرر مقدمة إلى البرمجة؟

أهمية الدراسة

منذ ما يربو على عشر سنوات، صرح جيم تكر، وهو أحد البارزين في علوم الحاسب، بأن هناك أزمة في تدريس علوم الحاسب (Tucker, 1996)، وقد ذكر بعض التحديات التي تواجه تدريس علوم الحاسب، وذكر منها: المناهج القديمة، وطرق التدريس غير المناسبة، وتدريب أعضاء هيئة التدريس. وحديثاً، أضاف بعض الباحثين عوامل أخرى كانعدام مهارات حل المشكلات لدى الطلاب، و معدلات الرسوب والانسحاب العالية من مقررات البرمجة، وقلة الطلاب الملتحقين بتخصص علوم الحاسب (Ma, Ferguson, Roper, & Wood, 2007; McBride, 2007). ولعقود من الزمن، حاول الباحثون في مجال تدريس علوم

الحاسب إلى تحسين تدريس علوم الحاسب عن طريق تحسين المناهج، وطرق التدريس (Tucker et al., 2003).

وستحاول هذه الدراسة تحسين الفهم الحالي للأسباب، والحلول الممكنة، لانخفاض أعداد الطلاب الملتحقين بتخصص علوم الحاسب؛ عن طريق استكشاف ما إذا كان تدريس الطلاب حل المشكلات غير الرياضية يؤثر بشكل ايجابي على فاعليتهم الذاتية في مجال البرمجة، وتحصيلهم الأكاديمي في مقرر المدخل إلى البرمجة. ومع زيادة فهم الباحثين بما يؤثر على الفاعلية الذاتية للطلاب في تخصص علوم الحاسب، فإن ذلك سيساعد المعلمين على تشجيع الطلاب على الالتحاق بذلك التخصص، وعدم الانسحاب منه. وقد لاحظ الباحثون العلاقة بين حل المشكلات وعلوم الحاسب، ولكن ستركز هذه الدراسة على علوم الحاسب كوسيلة تعليمية لحل المشكلات، وليس العكس.

وقد ناقش جين (Jin, 2008) تأثير تدريس حل المشكلات على أداء الطلاب الجامعيين، ولكنه لم يبحث في العلاقة بين تدريس حل المشكلات والفاعلية الذاتية للطلاب أو الاحتفاظ بالمعرفة. بينما بحث كومار (Kumar, 2008) تأثير حصص التدريب المتكررة (كأحد أنماط حل المشكلات) على الثقة الذاتية للطلاب الجامعيين. وستساعد نتائج الدراسة الحالية على تسهيل فهمنا للدور الذي يمكن أن يلعبه تدريس حل المشكلات في تحسين تدريس علوم الحاسب.

وتأتي هذه الدراسة منسجمة مع الدعوة إلى الاهتمام بتنمية مهارات التفكير الناقد لدى الطلبة (نجم، 2011) والعمل على الارتقاء بالعملية التعليمية من ثقافة الحفظ والتلقين إلى ثقافة الابداع والتفكير، وذلك من خلال محاولتها تحسين تعلم علوم الحاسب وبرمجته من خلال استخدام أسلوب حل المشكلات في التدريس، والعمل على توجيه نظر التربويين العاملين في حقل الحاسب من خبراء مناهج، ومؤلفي كتب، ومعلمين نحو إعداد مناهج دراسية واتباع طرائق تعلم تكون أكثر قدرة على تنمية مهارات الحوسبة.

وستساعد النتائج التي سنحصل عليها من هذه الدراسة على الحصول على إجابات حول تحسين تدريس علوم الحاسب، وتقليل نسبة الرسوب أو الانسحاب من تلك المقررات، إذ يعتبر إحجام الطلاب عن التخصص في علوم الحاسب مشكلة فعلية في هذا العصر الذي يتطلب الكثير من المختصين في هذا المجال. كما أن التعرف على الحلول الممكنة للمساعدة على تشجيع الطلاب وبقائهم في تخصص علوم الحاسب يعتبر إضافة ذات فائدة بالغة للمجال. إن نتائج هذه الدراسة يمكن أن تسهم بشكل فعال في نجاح الطلاب وتخرجهم وانخراطهم في سوق العمل.

كما يُتوقع أن تسهم هذه الدراسة في مساعدة أعضاء هيئة التدريس، ومعلمي مقررات البرمجة على تطوير أساليب واستراتيجيات تدريسية في المرحلتين الجامعية والثانوية قائمة على أساس

التعلم المتمركز حول المشكلة، لتدريس مهارات برمجة الحاسبات. كما يُتوقع أن تسهم نتائج هذه الدراسة في توجيه أنظار الباحثين وأعضاء هيئة التدريس والمعلمين إلى أهمية تضمين استراتيجيات تدريسية مبنية على نتائج البحث العلمي، وهو الشيء الذي أوصت به العديد من الدراسات السابقة (شريف، 2011؛ ياسن 2013). بعبارة أخرى، يمكن الاستفادة من نتائج الدراسة الحالية في تطوير طرائق التدريس بصفة عامة، وطرق تدريس برمجة الحاسبات بصفة خاصة في تقديم طرائق تسعى إلى إثارة تفكير المتعلم ونقل محور الارتكاز من التعليم إلى التعلم، ومن المعلم إلى المتعلم ومن الحفظ إلى التفكير والابتكار، وهذا ما تسعى إليه الاتجاهات المعاصرة في طرق التدريس، كما يعد مبرراً قوياً لإجراء مثل هذه الدراسة.

أسئلة وفروض الدراسة

أجابت هذه الدراسة عن السؤالين الرئيسيين التاليين:

1. ما أثر تدريس الطلاب مهارات حل المشكلات غير الرياضية ومهارات التفكير الناقد على الفاعلية الذاتية في مهام برمجة الحاسبات.
2. ما أثر تدريس الطلاب مهارات حل المشكلات غير الرياضية ومهارات التفكير الناقد على الأداء الأكاديمي في مهام برمجة الحاسبات.

وفيما يلي عرض لفروض الدراسة:

الفرضية الأولى: لا يوجد فروق ذات دلاله إحصائية عند مستوى 0.05 بين الطلاب الذين يتلقون تدريساً في حل المشكلات غير الرياضية ومهارات التفكير الناقد والطلاب الذين لم يتلقوا تدريساً في حل المشكلات ومهارات التفكير الناقد في مستوى الفاعلية الذاتية.

ويتفرع من هذه الفرضية أربع فرضيات فرعية.

الفرضية 1.1: لا يوجد فروق ذات دلاله إحصائية عند مستوى 0.05 بين الطلاب الذين يتلقون تدريساً في حل المشكلات غير الرياضية ومهارات التفكير الناقد والطلاب الذين لم يتلقوا تدريساً في حل المشكلات ومهارات التفكير الناقد في مستوى الفاعلية الذاتية **المتعلقة بقدرتهم على العمل باستقلالية، والاستمرار في العمل، بالرغم من الصعوبات.**

الفرضية 1.2: لا يوجد فروق ذات دلاله إحصائية عند مستوى 0.05 بين الطلاب الذين يتلقون تدريساً في حل المشكلات غير الرياضية ومهارات التفكير الناقد والطلاب الذين لم يتلقوا تدريساً في حل المشكلات ومهارات التفكير الناقد في مستوى الفاعلية الذاتية **المتعلقة بقدرتهم على أداء مهام برمجية معقدة.**

الفرضية 1.3: لا يوجد فروق ذات دلاله إحصائية عند مستوى 0.05 بين الطلاب الذين يتلقون تدريساً في حل المشكلات غير الرياضية ومهارات التفكير الناقد والطلاب الذين لم

يتلقوا تدريساً في حل المشكلات ومهارات التفكير الناقد في مستوى الفاعلية الذاتية **المتعلقة بقدرتهم على التنظيم الذاتي**.

الفرضية 1.4: لا يوجد فروق ذات دلاله إحصائية عند مستوى 0.05 بين الطلاب الذين يتلقون تدريساً في حل المشكلات غير الرياضية ومهارات التفكير الناقد والطلاب الذين لم يتلقوا تدريساً في حل المشكلات ومهارات التفكير الناقد في مستوى الفاعلية الذاتية **المتعلقة بقدرتهم على أداء مهام البرمجة البسيطة**.

الفرضية الثانية: لا يوجد فروق ذات دلاله إحصائية عند مستوى 0.05 بين الطلاب الذين يتلقون تدريساً في حل المشكلات غير الرياضية ومهارات التفكير الناقد والطلاب الذين لم يتلقوا تدريساً في حل المشكلات ومهارات التفكير الناقد في مستوى الفاعلية الذاتية في مستوى **الأداء الأكاديمي**.

حدود الدراسة:

اقتصرت هذه الدراسة على عينة من الطلاب الذكور الدارسين في مقرر برمجة الحاسبات (1)، والمسجلين خلال الفصل الدراسي الثاني من العام الجامعي 2013. وقد أجريت هذه الدراسة في جامعة الملك سعود في الرياض.

مصطلحات الدراسة

الحوسبة: هو مصطلح يشمل مفردات تستخدم لوصف التخصصات الخمسة المعروفة في مجال الحاسب: علوم الحاسب، وتقنية المعلومات، ونظم المعلومات، وهندسة الحاسب، وهندسة البرمجيات (Courte & Bishiob-Clarck, 2009).

حل المشكلات غير الرياضية: تعرف بأنها عمليات حل مشكلات عامة ليست متعلقة على وجه التحديد بعملية حل أو فهم مسائل الرياضيات الكتابية. وتشمل تلك العمليات استراتيجيات يمكن تطبيقها على نطاق عام من المشاكل، ومن أمثلتها الاحجيات المنطقية، والجدولة (Whimbley & Lochhead, 1999).

حل المشكلات: ويعرف بأنه العمليات الذهنية عالية المستوى التي تستخدم لأجل التوصل إلى حلول لمواقف معطاة (Maddery, 2011؛ أحمد، 2011).

أسلوب التفكير المسموع: ويعرف بأنه أسلوب يتم من خلاله حل المشكلات عن طريق تصور الطلاب كل خطوة من العمليات الذهنية بأنفسهم أو مع قرين. ومع التدريب المستمر، يمكن تطبيق اسلوب التفكير المسموع دون الحاجة إلى التصور المسموع للعمليات الذهنية ويكتفي المتعلم بالتفكير ذاتياً في تلك العمليات (Whimbley & Lochhard, 1999).

الفاعلية الذاتية: يعرفها بندورا (Bandora, 1994) بأنها اعتقاد الفرد بقدرته على القيام بمهمة معطاة.

الفاعلية الذاتية في برمجة الحاسب: وتعرف بأنها اعتقاد الفرد بقدرته على القيام بمهمات برمجية.

برمجة الحاسب: ويقصد بها الوسائل التي يمكن من خلالها إيصال التعليمات المرتبة وفق تسلسل محدد إلى الحاسب الآلي، والتي تجعل من الحاسب آلة تستطيع القيام بالمهام أسرع وأفضل من الإنسان، وذلك بناء على التعليمات المعطاة له من قبل المبرمج (جامع، 2012). ويقصد بها في هذه الدراسة الأوامر والأكواد، التي تسهل للمبرمج عملية التعامل مع الكمبيوتر، لكي يتمكن من تنفيذ المهام التي يرغب في تنفيذها من خلال البرمجة.

الإطار النظري والدراسات السابقة

درس الأكاديميون والمختصون الانخفاض في أعداد الطلاب المنخرطين بي برامج الحوسبة بشكل عام، وهو الشيء الذي لاحظه الباحث في دراسة استطلاعية أجراها على طلاب الجامعة. وأظهرت نتائج الدراسات أن المشكلة تنبع من استقطاب الطلاب في المراحل الأولى (Hart et al., 2008; Van Sickle, 2008) وتستمر المشكلة في المحاولة على الإبقاء على الأعداد المحدودة التي بدأت في البرنامج الأكاديمي (Barker, McDowel, & Kalahar, 2009; Wilson, 2008).ومن الأسباب الأساسية لانخفاض الانخراط في برامج الحوسبة، والتي أظهرتها الدراسات السابقة: الاعتقادات الخطأ حول حقل الحوسبة (Beaubouef & McDowel, 2008; Rafiermehr, 2008)، والصور الاجتماعية النمطية السلبية حول المنخرطين في هذا الحقل (Adya, 2008; Edmondson, 2008)، وانعدام الفاعلية الذاتية (Cohoon et al., 2008; Norries et al., 2008). وقد أظهر تحليل للطلاب المنسحبين في سنتهم الأولى أن من أهم التحديات التي تواجه الطلاب هو انعدام مهارات حل المشكلات لديهم (Beaubouef et al., 2001, Jin, 2008; Kumar, 2008).

أظهرت الدراسات أن هناك عدداً من المعتقدات الخطأ حول تعلم علوم الحاسب، ومن تلك المعتقدات أن هناك حاجة إلى درجة عالية من المعرفة بالرياضيات من أجل النجاح في دراسة البرمجة (Colley et al., 1996). وعلى الرغم من ذلك فقد أظهرت الدراسات أنه لا علاقة بين مستوى المعرفة في الرياضيات والانخراط في تخصصات الحوسبة. فقد وجد كولي وزملاؤه (Colley et al., 1996) أن المفهوم السائد بأن الرياضيات مهارة متطلبة للنجاح في تخصصات الحوسبة قد أدى إلى انخفاض رغبة الطلاب وإقبالهم على الحوسبة. أما سكراج وسميث (Scragg & Smith, 1998) فقد درسا القلق من الرياضيات كأحد مسببات انسحاب الطلاب من مقررات البرمجة، ووجدا أنه بينما يدرك الطلاب أهمية الرياضيات في دراسة البرمجة، إلا أن ذلك لم يكن مسبباً لانسحابهم من المقررات. وبالمثل، وجد كل من نوتا

وإبيرسون (Nauta & Epperson, 2003) بأن القدرات الرياضية ليس لها تأثير ذو دلالة إحصائية على قدرات الطالب في الحوسبة. ويظهر من هذه الدراسات أنه لا علاقة مباشرة بين القدرات الرياضية للطالب والنجاح في مقررات علوم الحاسب.

ومن المعتقدات الخطأ حول الحوسبة، أنه ليس لها علاقة بمشاكل ذات أهمية في قطاعات أخرى. وتدعم الممارسات التدريسية الحالية هذا الفهم، بتركيزها فقط على لغات البرمجة وتطوير البرمجيات في مقررات الحاسب الأولية (Heart et al., 2008; Hazzan, Gal-Ezer, & Blum, 2008). وقد لاحظ الن وكولزر (Allan & Coleasar, 1997) أن الطلاب في مقررات علوم الحاسب الأساسية أظهروا تركيزاً فقط على إنهاء المهمة البرمجية المعطاة دون التفكير في الصورة الأكبر، وأن المهارات التي يتعلمونها يمكن تطبيقها في ميادين شتى، أو حتى تطبيقها في مجال الحوسبة ككل. وقد دعت العديد من الدراسات إلى تغيير مناهج علوم الحاسب لتغيير هذا الصورة عن مجال الحوسبة ككل.

وقد ناقش كل من كلاوي وشندرمان (Klwe & Shnedeiderman, 2005) أهمية إحداث نقلة في مقررات علوم الحاسب لمعالجة استخدام الحاسب في حل المشاكل الاجتماعية، وهو الشيء الذي تسعى إليه هذه الدراسة من خلال تدريس مهارات حل المشكلات غير الرياضية من خلال مقررات البرمجة. ويرى بيكر وزملاؤه (Baker et al., 2007) أن نظرة الطلاب إلى المجالات الهندسية والحاسوبية، وأنها ليست ذات علاقة بمشاكل المجتمع المباشرة؛ تنفرهم من الالتحاق بتلك التخصصات. أما راو (Rao, 2006) فيؤكد على الحاجة إلى أن يقوم المعلمون في حقل علوم الحاسب إلى إحداث نقلة في إستراتيجياتهم بحيث يركزون على التطبيقات في المجالات التي تهم الطلاب بدلاً من التركيز على النظريات المجردة. ويؤيده في ذلك فان سيكل (Van Sickle, 2008) إذ يوصي بإحداث نقلة في مقررات علوم الحاسب وذلك بأن تركز على التخصصات والتطبيقات التي تحتاجها الصناعة بدلاً من النظريات التي تقدمها معظم البرامج الأكاديمية في علوم الحاسب. أما جوسيف (Joseph, 2008) فيرى أهمية تضمين برامج الحاسب الآلي مقررات التدريب الميداني، لإن الدراسات أظهرت أن ذلك يؤثر بصورة ايجابية على اختيار الطلاب لتخصصات الحوسبة. وأخيراً، يقترح كلابهولز (Klappholz, 2009) تضمين مشاريع برمجية حقيقية في مقررات علوم الحاسب، ويقوم الطلاب بصيانة هذه المشاريع وتعديلها ومن ثم استخدامها في القسم الأكاديمي أو لصالح مؤسسة تجارية، وهذا يوفر للطلاب الفرصة للعمل مع عميل حقيقي، وليس افتراضياً. ونستخلص من الدراسات السابقة أن مناهج علوم الحاسب الحالية تعتمد على أمثلة افتراضية أو تخيلية معزولة عن واقع سوق العمل، وأنه بالإمكان مساعدة مجال علوم الحاسب بجعله وثيق الصلة بحياة الطلاب بإحداث نقلة في مقررات علوم الحاسب وتضمينها مشاكل من أرض الواقع، وربما يؤدي ذلك إلى إقبال الطلاب على هذا التخصص.

الفاعلية الذاتية في مجال الحاسب

يعرف باندورا (Bandora, 1994) الفاعلية الذاتية بأنها اعتقاد المرء بقدرته إنهاء مهمة معطاة بنجاح. ويضيف بندورا بأن الفاعلية الذاتية تتأثر بأربعة عوامل رئيسية: الخبرات الشخصية السابقة، ومراقبة خبرات الآخرين، والحوافز الخارجية الفعالة، واستجابة الفرد العاطفية للمهمة المعطاة. وتبدأ الفاعلية الذاتية بالتطور لدى الإنسان منذ نعومة أظفاره، عندما يبدأ في استكشاف العالم المحيط به، وتستمر معه خلال الأطوار المختلفة لحياته. وكلما تقدم الطفل في النمو، يصبح الأباء والأقران والمدارس عوامل تؤثر على تطور الفاعلية لذاتية. ويعتقد كل من بنترتش وشنك (Pintrich & Schunk, 1996) أن الفاعلية الذاتية للطالب تتناقص حتى يبلغ المرحلة المتوسطة. ويشير كل من ليو وزملاؤه (Liu, Hseih, Cho, & Schallert, 2006) إلى أن الفاعلية الذاتية في مجال العلوم والتقنية تبدأ في الانخفاض ابتداءً من سن الحادية عشر. ويشير تانج وزملاؤه (Tang et al., 2008) إلى أن الطالب يكّون اعتقادات راسخة حول فاعليته الذاتية في عمر الرابعة عشر. وقد وجد كل من ناوتا وابيرسون (Nauta & Epperson, 2003) أنه يمكن التأثير على الفاعلية الذاتية للطالب بشكل ايجابي في المرحلة الثانوية والجامعية، وخاصة إن كان تحصيل الطالب ودافعيته تدعمان بقاءه في التخصص. ويرى مايال (Mayall, 2008) بأنه بمكن التأثير بشكل ايجابي على الفاعلية الذاتية لطلاب الجامعة، ومن ثم اتجاهاتهم نحو قدراتهم واختيارهم للحوسبة كتخصص.

وقد لاحظ بيكويذ وزملاؤه (Beckwith et al., 2006) أن التجارب الأولى مع الحوسبة تساعد على تكوين الأساس للفاعلية الذاتية في مجال الحوسبة، وأن الفشل المبكر، أو حتى تصور الفشل يؤدي إلى تأسيس فاعلية ذاتية منخفضة في المستقبل، على الرغم من إمكانية النجاح في المستقبل.

وتوصي جمعية معلمي علوم الحاسب (http://www.csta.acm.org) بتطوير وتنفيذ مقررات للحاسب الآلي تبدأ من ما قبل المرحلة الابتدائية وتستمر حتى نهاية المرحلة الثانوية. وعادة تبدأ تلك المقررات بتعريض الطالب للحاسبات عن طريق دمج المفاهيم الحاسوبية في المقررات الدراسة الأخرى، ويكون هذا في المرحلة الابتدائية والمتوسطة، وذلك من أجل تنمية ثقافة الحاسب لدى الطالب، وإعطائه فرصة للتفاعل الايجابي مع الحاسب. أما في المرحلة الثانوية، فتوصي الجمعية بأن يتم تصميم مقررات الحوسبة لتشجيع الطلاب على تطوير مهارات البرمجة وفهم النظريات الأساسية في علوم الحاسب (DeClue, 2008). وقد وجد كادي وتيرل (Cady & Terrell, 2007) أن دمج التقنية في مقررات العلوم قد ساعد على دعم الفاعلية الذاتية للطلاب فيما يتعلق باستخدام التقنية، أكثر من مجرد تدريس التقنية بدون

استخدامها في سياق محدد. وهذا الأسلوب يطابق ما أوصى به ديلكلوي (DelClue, 2008) في مجال تعليم الحاسب في المرحلة الابتدائية. ويتفق رافيمهر (Rafieymehr, 2008) مع الدراسات السابقة التي وجدت أن التعرض المبكر للحاسب يعتبر حجر أساس في بناء الفاعلية الذاتية والاهتمام في تخصصات الحوسبة.

وبما أن الفاعلية الذاتية خاضعة بشكل كبير لنشاط محدد، فإن كاسيدي وايكوس (Cassidy & Eachus, 2002) يؤكدان على أن الفاعلية الذاتية للفرد تختلف باختلاف المجال، وبالتالي، فقد يكون الشخص على ثقة عالية في قدراته الرياضية، ولكنه يحمل قدراً ضئيلاً من الفاعلية الذاتية في مجال الحوسبة. ويؤيد كل من دوني (Downey, 2006) وليو وزملاؤه (Liu et al., 2006) هذه الفكرة، ويضيفون بأن الفاعلية الذاتية قد تختلف في ذات المجال مثل مجال الحوسبة، فالفاعلية الذاتية في مجال الحوسبة قد تختلف باختلاف المهمة المطلوب انجازها. ففي هذه الحالة قد يظهر المرء قدراً عالياً من الفاعلية الذاتية في مجال الاستخدام العام للحاسب الآلي، ولكنه قد يعاني من قدر أقل من الفاعلية الذاتية عندما يتعلق الأمر بحل مسألة برمجية.

ويوصي ماديقان وآخرون (Madigan et al., 2007) بأن يجد معلمو علوم الحاسب الآلي طرقاً لبناء الثقة تدريجياً لدى الطلاب من خلال مقررات علوم الحاسب، ولكنهم لا يقدمون اقتراحات حول الاستراتيجيات الفعالة التي يمكن اتباعها لبناء تلك الثقة. ويقترح كواد (Quade, 2003) بأن يتوقع المعلمون من طلابهم قدراً عالياً من الفاعلية لأجل تحفيزهم للصمود في وجه انخفاض الفاعلية الذاتية لديهم. وقد لاحظ كواد بأن النجاح في حل المشكلات كان مكوناً هاماً في بناء الفاعلية الذاتية في مجال الحاسب لدى الطلاب.

الخبرة السابقة في مجال الحاسب وعلاقتها بالفاعلية الذاتية

وجد كل من بالو وهوقوينارد (Ballou & Huguenard, 2008) أن الطلاب يصدرون أحكاماً حول إمكانيات نجاحهم في مقررات الحاسب بناء على تصوراتهم حيال خبراتهم السابقة، ومن ثم فإنه يجدر تطوير استراتيجيات تعوض النقص في الخبرات السابقة لدى الطلاب. وإحدى الاستراتيجيات المستخدمة، هي تدريس مفاهيم البرمجة عبر القصص في بيئات البرمجة المرئية، مما يقلل الاعتماد على الخبرات السابقة في المقررات الأولية (Moskal et al., 2004). أما ساند وزملاؤه (Sands et al., 2008) فقد طوروا مقرراً تمهيديا في علوم الحاسب يعتمد على تدريس أحد تطبيقات الحاسب وهو الفلاش. ويمكن استخدام الفلاش لأجل تصميم العروض متعددة الوسائط، ولكنه يحوي أيضاً لغة برمجة عالية المستوى للمستخدمين المحترفين. وقد كان الهدف من ذلك المقرر؛ خلق نوع من الاهتمام والدافعية لدى الطلاب إلى استخدام التطبيق، ثم الانتقال تدريجياً إلى استخدام لغة البرمجة التي يحويها الفلاش متى ما أدرك الطلاب فائدة البرنامج.

وقد قدم كل من مولينز وويتفيلد وكونلون (Mullins, Whitfield, & Conlon, 2008) مقرراً دراسياُ في البرمجة معمتداً على لغة "أليس" وهي لغة برمجة ثلاثية الابعاد تعتمد على السحب والإفلات، وقد كان هذا المقرر الأول من بين ثلاثة مقررات في البرمجة. وعلى الرغم من العقبات التي واجهها الباحثون في تدريس لغة البرمجة؛ إلا أن الطلاب تعلموا مبادئ برمجية أساسية من خلال التجربة المرئية، وقد أظهروا قدراً عالياً من الأداء مقارنة بزملائهم الذين لم يتعلموا مبادئ البرمجة من خلال "اليس". كما أظهرت نتائج الدراسة انخفاضاً ملحوظا في عدد الطلاب الين ينسحبون من مقررات البرمجة الثلاثة. أما سيفيلوتي ولاوجيل (Sivilotti & Laugel, 2008) فقد وظفوا لغة "سكراتش" من أجل أن يوفروا لطلاب المرحلة المتوسطة طريقة سهلة ومشوقة لتعلم مفاهيم البرمجة من خلال الوسائط المتعددة. وقد استخدم آلبو وزملاؤه (Al–Bow et al., 2008) بيئة "قرين فوت" التي تعتمد على السحب والإفلات والشبيهة بـ"أليس" من أجل تدريس المدخل إلى البرمجة بلغة جافا لطلاب المرحلة الثانوية الذين ليس لديهم خبرة سابقة في البرمجة. وقد أشار الطلاب إلى أنهم قد اكتسبوا فهماً عميقاً لمفاهيم البرمجة من خلال بيئة "قرين فوت".

حل المشكلات والتفكير الناقد

تعتبر مهارات حل المشكلات مكوناً مهماً للنجاح في علوم الحاسب، وهذا ما تحاول الدراسة الحالية اثباته من خلال إجراء دراسة شبه تجريبية على عينة من الطلاب السجلين في مقرر مقدمة للبرمجة. ويعترف بهذه الحقيقة معظم الممارسين والطلاب الذين أتموا السنة الأولى من دراستهم في تخصص علوم الحاسب؛ ولكن لا يتم الإشارة إلى تلك الحقيقة في كتب علوم الحاسب أو في غرفة الصف (Biggers et al., 2008; DeClue, 2008). وقد وجد كل من قولد وريمير(Goold & Rimmer, 2000) أن القدرة على حل المشكلات من أهم المؤشرات لنجاح الطلاب في السنة الأولى في تخصص علوم الحاسب.

وقد درس تشو (Cho, 1995) استخدام لغة البرمجة الرسومية "لوقورايتر" (LogoWriter) كأحد أساليب تدريس البرمجة في مقرر المدخل إلى البرمجة في تخصص علوم الحاسب؛ وقد خلصت الدراسة إلى أن التركيز على مهارات حل المشكلات في سياق البرمجة قد ساعد على تخفيف نسبة القلق لدى الطلاب، وزيادة مستوى الثقة لديهم فيما يتعلق بقدراتهم التقنية. ويرى كولي وزملاؤه (Colley et al., 1996) وراو (Rao, 2006) ضرورة إعادة تعريف الحوسبة كمجال يعتمد على حل المشكلات بدلاً من كونه معتمداً على الرياضيات، وأن ذلك ربما يؤدي إلى رفع أداء الطلاب واهتمامهم بالمجال.

وقد بحث كل من جبسون وأو كلي (Gibson & O'Kelly) وأولسن (Olsen, 2005) عدداً من الطرق لاستخدام الحاسب كأداة لحل المشكلات لغير المختصين في مجال الحاسب،

وتدريس طلاب علوم الحاسب أسلوباً مكوناً من أربع خطوات لحل المشكلات؛ ويمكن تلخيص هذا الأسلوب بأنه يعتمد على تلخيص المشكلة، ثم تكوين خطة لحل المشكلة، ثم تطبيق الخطة، فالتحقق من الحل. وللمتخصصين في علوم الحاسب، فإن خطوات هذا الأسلوب تكون: تحديد مدخلات ومخرجات البرنامج، ثم تصميم البرنامج باستخدام مخططات الانسياب والألقورثمات، ثم كتابة البرنامج، وأخيراً تشخيص الأخطاء في البرنامج وتجريبه. وقد وجد الباحثون أن هذا أسلوب فعال لتدريس البرمجة.

ويوصي ايستمان (Eastman, 2003) على وجه التحديد بالتركيز على تجزئة المشكلة إلى حقائق منفصلة كخطوة أولى في سلسلة حل المشكلة (فهم المشكلة)، ومن خبرته يرى ايستمان أن الطلاب يتبعون خطوات حل المشكلة بدون فهم المشكلة أو الخطوات التي اتبعوها لحلها، ويؤكد على أهمية فهم المشكلة كخطوة أولى للتوصل إلى حل لها. ويوصي كلاً من ليا وونق (Lia & Wong, 2007) باستخدام الرسومات البيانية لمساعدة الطلاب في مقررات البرمجة على تحليل المشكلة وتجزئتها. ويرى جين (Jin, 2008) أن مساعدة المدرس للطلاب في تحليل المشكلة في المهام التي يقوم بها الطلاب في مقررات البرمجة قد ساعد الطلاب على التعرف على المدخلات، والعمليات، والمخرجات مما ساعد الطلاب على فهم المسألة البرمجية بشكل أسرع. ويؤكد علي (Ali, 2005) على أهمية تدريس حل المشكلات والتفكير الناقد كجزء من مقررات البرمجة الأولية في تخصصات الحاسب وهذا ما سيقوم به الباحث في هذه الدراسة، مع اختلاف لغة البرمجة المستخدمة. ففي الدراسة الحالية يستخدم الطلاب لغتي سي وجافا لأجل التعرف على المفاهيم الأساسية في البرمجة. ويؤكد ذلك اولسن (Olsen, 2005) إذ يرى عدم التركيز على الأخطاء البرمجية، والمنهجية في بداية تدريس البرمجة، وإنما يتم التركيز على خطوات حل المشكلة باستخدام لغة بسيطة.

بالإضافة إلى ما سبق، يوصي الباحثون باستخدام المجموعات في تدريس مقررات البرمجة وذلك في المراحل الأولية من المقرر (مرحلة حل المشكلات) (Ali, 2005; Gibson & O'kelly, 2005). ويؤكد على ذلك كل من بوليمود وولز (Pulimood & Wolz, 2008)، إذ يريان ضرورة التركيز ليس فقط على مهارات حل المشكلات، وإنما مهارات حل المشكلات داخل المجموعات الطلابية عند تدريس مقررات البرمجة، إذ يعتبر ذلك محاكاة لبيئة العمل التي سيعمل بها الطالب بعد تخرجه. ويتفق مع ما سبق كل من فالكنر وبالمر (Falkner & Palmer, 2009). إذ يوصون بتضمين محاضرات في حل المشكلات، حيث يمارس الطلاب نشاطات حل المشكلات، يتبعها نشاطات تعاونية في حل المشكلات. ويرى فالكنر وبالمر أن جانب العمل الجامعي في حل المشكلات مهم للتطبيق الفعلي لما يتعلمه الطالب في العالم الخارجي أثناء عمله، وأنه لا يمكن إغفال هذا الجانب. أما هانكس وبراندت (Hanks & Brandet, 2009) فقد استخدما أسلوب التعلم مع القرين لأجل تدريس مهارات

البرمجة، ووجدا أن الطلاب يمضون وقتاً قصيراً جداً في مناقشة المسألة البرمجية وتحديد طبيعتها والخطوات اللازمة لحلها، ويوصيان المدرسين بضرورة مساعدة الطلاب من أجل التعرف على أهمية المرحلة الأولى من البرمجة والمتمثلة في خطوات حل المشكلة. وفي الدراسة الحالية، سيتعلم الطلاب أسلوب حل المشكلات غير الرياضية، ويستخدمونه في داخل مجموعات طلابية تعلمية.

ويؤكد ليمير (Lemire, 2002) على عدم وجود دراسات تدعم الفكرة القائلة بأن مهارات حل المشكلات التي يتعلمها الطالب في أحد المجالات، كالرياضيات مثلاً، يمكن استخدامها في مجالات أخرى دون الحاجة إلى تدريس مهارات حل المشكلات في ذلك المجال. ويرى ليمير أنه حتى تصبح مهارات حل المشكلات فعالة، فإنه يجب تدريسها من خلال المجال الذي ستستخدم فيه، وفي الدراسة الحالية، سيتم تدريس مهارات حل المشكلات في سياق تعلم لغة البرمجة. ويعكس ما توصل إليه ليمير النتائج التي توصل إليها بالمبو (Balumbo, 1990) والذي راجع عدداً كبيراً من الدراسات في مجال حل المشكلات، ولم يجد ما يثبت إمكانية انتقال مهارات حل المشكلات من مجال ما، كالرياضيات، إلى مجال آخر، كعلوم الحاسب.

منهجية الدراسة

اتبعت هذه الدراسة المنهج شبه التجريبي المتضمن اختباراً قبلياً وبعدياً ومجموعتين: إحداهما تجريبية، والأخرى ضابطة. وقد أجريت الدراسة على شعبتين من مقرر مقدمة إلى البرمجة خلال الفصل الدراسي الثاني من العام 2003. وقد أتم الطلاب في كلا المجموعتين مقياس الفاعلية الذاتية في برمجة الحاسبات قبل الشروع في تدريس المقرر. وقد تم تدريس المجموعة الضابطة باستخدام أسلوب التدريس الذي يتبعه أستاذ المقرر عادة في تدريسه، دون تقديم تدريب خاص حول حل المشكلات. أما المجموعة التجريبية فقد تلقت تدريساً إضافياً في حل المشكلات في بداية الفصل الدراسي وتم التركيز على مهارات التفكير الناقد اللازمة لحل المسائل البرمجية. وفي منتصف الفصل الدراسي، ونهايته تم تقديم مقياس الفاعلية الذاتية في برمجة الحاسبات مرة أخرى.

أدوات الدراسة

قبل الشروع في الدراسة، طُلب من الطلاب في المجموعتين الضابطة والتجريبية إتمام مقياس الفاعلية الذاتية في برمجة الحاسبات والذي أعده كل من رامالينقام وويدنوبيك (Ramalingam & Wiedenbeck, 1998) وذلك بعد تعريبه وعرضه على خمسة أساتذة في قسم المناهج وطرق التدريس للتأكد من سلامة الترجمة. ويستخدم هذا المقياس لقياس الفاعلية الذاتية في برمجة الحاسبات لدى الطلاب، ويحوي أربعة محاور رئيسية: الاستقلالية والمثابرة، والقدرة على أداء مهام البرمجة المعقدة، والتنظيم الذاتي، والقدرة على أداء مهام

البرمجة البسيطة. ويبغ معامل الثبات للمقياس ككل 0.98، ومعامل الثبات للمحاور الأربعة مرتبة كما يلي: 0.94 و0.94 و0.86 و0.93 وتعتبر هذه القيم مناسبة لاستخدام المقياس في هذه الدراسة. ويحوي المقياس 31 عبارة يستجيب لها الطلاب على مقياس لايكرت سباعي. وقد استجاب الطلاب للمقيس في بداية الفصل الدراسي، ومنتصفه، ونهايته.

إجراءات الدراسة

خلال الأسبوع الأول من الفصل الدراسي؛ تم إعطاء الطلاب في المجموعتين التجريبية والضابطة مقياس الفاعلية الذاتية في برمجة الحسبات، وتم تقديم الاختبار القبلي للطلاب. وطلب من الطلاب في المجموعة التجريبية مطالعة برنامج تعليمي إلكتروني مكون من خمسة دروس في حل المشكلات موجود على نظام إدارة التعلم Black Board، وقد طلب من الطلاب في المجموعة التجريبية أخذ اختبار إلكتروني حول حل المشكلات وذلك للتأكد من إتمامهم جميعاً البرنامج التعليمي ي حل المشكلات.

ومن خلال البرنامج التعليمي؛ تم تدريب الطلاب على أسلوب لحل المشكلات اسمه: التفكير بصوت عال، والذي يساعد الطلاب على القيام بحل المشكلات شفوياً مع زميل مستمع كخطوة أولية تقود إلى القدرة على حل المشكلات بدون الحاجة إلى النطق. وقد طور هذا النظام كل من ويمبلي ولوكهيد (Whimpley & Lochhead, 1999) وأثبتا من خلال عدد من الدراسات فاعليته في تطوير مهارات حل المشكلات. كما تم تدريس الطلاب أسلوب التعرف على المدخلات والعمليات والمخرجات التي تقدم في المسائل البرمجية. من ثم تحويل تلك المعلومات إلى حلول باستخدام لغة البرمجة. وتعتبر هذه الخطوات في حل المشكلات متسقة مع ما اقترحه الباحثون السابقون في مجال حل المشكلات في مجال البرمجة (Jin, 2008).

وعند منتصف الفصل الدراسي، طلب من الطلاب في كلا المجموعتين إكمال مقياس الفاعلية الذاتية في برمجة الحاسبات، كما تم حث الطلاب في المجموعة التجريبية على مراجعة البرنامج التعليمي في حل المشكلات والذي تلقوه في بداية الفصل الدراسي. كما أتم الطلاب اختبار منتصف الفصل الدراسي كما هو مخطط لهم في مفردات المقرر.

وفي نهاية الفصل الدراسي، طلب من الطلاب إتمام مقياس الفاعلية الذاتية في برمجة الحاسبات، كما طلب من الطلاب إتمام الاختبار البعدي.

النتائج

سيتم استعراض النتائج وفقاً لفرضيات الدراسة. وقد كانت الفرضية الأولى: سيُظهر الطلاب الذين يتلقون تدريساً في حل المشكلات غير الرياضية ومهارات التفكير الناقد قدراً أعلى ودالاً

إحصائياً من الفاعلية الذاتية في مهام برمجة الحاسبات، من الطلاب الذين لم يتلقوا تدريساً في حل المشكلات غير الرياضية ومهارات التفكير الناقد.

ويتفرع من هذه الفرضية أربع فرضيات فرعية.

الفرضية 1.1: لا يوجد فروق ذات دلاله إحصائية عند مستوى 0.05 بين الطلاب الذين يتلقون تدريساً في حل المشكلات غير الرياضية ومهارات التفكير الناقد والطلاب الذين لم يتلقوا تدريساً في حل المشكلات ومهارات التفكير الناقد في مستوى الفاعلية الذاتية **المتعلقة بقدرتهم على العمل باستقلالية، والاستمرار في العمل، بالرغم من الصعوبات.**

يعرض الجدول رقم 1 المتوسطات والانحرافات المعيارية لاستجابات أفراد المجموعتين التجريبية والضابطة للمحور الأول من مقياس الفاعلية الذاتية في البرمجة والذي يركز على قدرة الطلاب على العمل باستقلالية والاستمرار في العمل بالرغم من الصعوبات.

جدول 1

المتوسطات والانحرافات المعيارية لاستجابات أفراد العينة للمحور الأول من مقياس الفاعلية الذاتية في البرمجة.

المجموعة	التطبيق القبلي			التطبيق البعدي		
	ن	م	إ.م.	ن	م	إ.م.
التجريبية	25	3.21	0.98	25	6.32	1.08
الضابطة	23	3.31	1.05	23	4.47	1.11

ويلاحظ من الجدول رقم 1 تحسن في الفاعلية الذاتية للطلاب من التطبيق القبلي إلى التطبيق البعدي. كما يلاحظ أن أفراد المجموعة التجريبية قد حققوا متوسط درجات أعلى من أفراد المجموعة الضابطة في التطبيق البعدي للأداة. وللتعرف على مستوى دلالة تلك الفروق، فقد تم استخدام اختبار التباين الأحادي.

جدول 2.

اختبار التباين الأحادي لمستوى دلالة الفروق بين متوسطات التطبيق القبلي والبعدي للمحور الأول من مقياس الفاعلية الذاتية في البرمجة.

المصدر	مجموع المربعات	درجات الحرية *df*	متوسط المربعات	قيمة ف	مستوى الدلالة *p*	حجم التأثير η^2
بين المجموعات	0.235	1	0.235	0.179	0.001	0.5
الخطأ	25.434	45	0725			

ويلاحظ من الجدول رقم 2 أن مستوى الدلالة للفروق بين متوسطات استجابات أفراد العينة للمحور الأول من مقياس الفاعلية الذاتية دال احصائياً. وعليه فإنه يتم رفض الفرضية الأولى

ويمكن القول بأن الطلاب الذين تلقوا تدريساً في حل المشكلات غير الرياضية ومهارات التفكير الناقد قد حققوا قدراً أعلى ودالاً احصائياً من الفاعلية الذاتية المتعلقة بقدرتهم على العمل باستقلالية والاستمرار في العمل بالرغم من الصعوبات أكثر من الطلاب الذين لم يتلقوا تدريساً في حل المشكلات ومهارات التفكير الناقد. كما أن حجم التأثير لهذا الأثر متوسط.

الفرضية 1.2: لا يوجد فروق ذات دلاله إحصائية عند مستوى 0.05 بين الطلاب الذين يتلقون تدريساً في حل المشكلات غير الرياضية ومهارات التفكير الناقد والطلاب الذين لم يتلقوا تدريساً في حل المشكلات ومهارات التفكير الناقد في مستوى الفاعلية الذاتية المتعلقة بقدرتهم على أداء مهام برمجية معقدة.

يعرض الجدول رقم 3 المتوسطات والانحرافات المعيارية لاستجابات أفراد المجموعتين التجريبية والضابطة للمحور الثاني من مقياس الفاعلية الذاتية في البرمجة والذي يركز على قدرة الطلاب على إجراء عمليات البرمجة المعقدة، مثل تصميم وفهم وتعديل والعثور على الأخطاء البرمجية في البرامج المعقدة، واستخدام الأكواد البرمجية التي كتبها الاخرون.

جدول 3.

المتوسطات والانحرافات المعيارية لاستجابات أفراد العينة للمحور الثاني من مقياس الفاعلية الذاتية في البرمجة.

المجموعة	التطبيق القبلي			التطبيق البعدي		
	ن	م	إ.م.	ن	م	إ.م.
التجريبية	25	2.47	1.247	25	5.87	1.035
الضابطة	23	3.02	0.357	23	4.24	0.745

ويلاحظ من الجدول رقم 3 تحسن في الفاعلية الذاتية للطلاب من التطبيق القبلي إلى التطبيق البعدي. كما يلاحظ أن أفراد المجموعة التجريبية قد حققوا متوسط درجات أعلى من أفراد المجموعة الضابطة في التطبيق البعدي للأداة. وللتعرف على مستوى دلالة تلك الفروق، فقد تم استخدام اختبار التباين الأحادي.

جدول 4.

اختبار التباين الأحادي لمستوى دلالة الفروق بين متوسطات التطبيق القبلي والبعدي للمحور الثاني من مقياس الفاعلية الذاتية في البرمجة.

المصدر	مجموع المربعات	درجات الحرية *df*	متوسط المربعات	قيمة ف	مستوى الدلالة *p*	حجم التأثير η^2
بين المجموعات	0.487	1	0.487	0.687	0.013	0.4
الخطأ	30.254	45	0.874			

ويلاحظ من الجدول رقم 4 أن مستوى الدلالة للفروق بين متوسطات استجابات أفراد العينة للمحور الثاني من مقياس الفاعلية الذاتية دال احصائياً. وعليه فإنه يتم رفض الفرضية الفرعية الثانية ويمكن القول بأن الطلاب الذين تلقوا تدريساً في حل المشكلات غير الرياضية ومهارات التفكير الناقد قد أظهروا قدراً أعلى ودالاً إحصائياً من الفاعلية الذاتية المتعلقة بقدرتهم على أداء مهام برمجية معقدة أكثر من الطلاب الذين لم يتلقوا تدريساً في حل المشكلات ومهارات التفكير الناقد. كما أن حجم التأثير لهذا الأثر قريب من المتوسط.

الفرضية 1.3: لا يوجد فروق ذات دلاله إحصائية عند مستوى 0.05 بين الطلاب الذين يتلقون تدريساً في حل المشكلات غير الرياضية ومهارات التفكير الناقد والطلاب الذين لم يتلقوا تدريساً في حل المشكلات ومهارات التفكير الناقد في مستوى الفاعلية الذاتية المتعلقة بقدرتهم على التنظيم الذاتي.

يعرض الجدول رقم 5 المتوسطات والانحرافات المعيارية لاستجابات أفراد المجموعتين التجريبية والضابطة للمحور الثالث من مقياس الفاعلية الذاتية في البرمجة والذي يركز على قدرة الطلاب على التحكم وضبط النفس من أجل الوصول إلى الهدف المراد، ويشمل ذلك القدرة على الاستمرار في العمل عند التعرض للملهيات والقدرة على العمل تحت ضغوط الوقت.

جدول 5

المتوسطات والانحرافات المعيارية لاستجابات أفراد العينة للمحور الثالث من مقياس الفاعلية الذاتية في البرمجة.

المجموعة	التطبيق القبلي			التطبيق البعدي		
	ن	م	إ.م.	ن	م	إ.م.
التجريبية	25	4.27	1.487	25	6.25	1.254
الضابطة	23	4.38	1.028	23	4.89	1.842

ويلاحظ من الجدول رقم 5 تحسن في الفاعلية الذاتية للطلاب من التطبيق القبلي إلى التطبيق البعدي. كما يلاحظ أن أفراد المجموعة التجريبية قد حققوا متوسط درجات أعلى من أفراد المجموعة الضابطة في التطبيق البعدي للأداة. وللتعرف على مستوى دلالة تلك الفروق، فقد تم استخدام اختبار التباين الأحادي.

جدول 6.

اختبار التباين الأحادي لمستوى دلالة الفروق بين متوسطات التطبيق القبلي والبعدي للمحور الثالث من مقياس الفاعلية الذاتية في البرمجة.

المصدر	مجموع المربعات	درجات الحرية *df*	متوسط المربعات	قيمة ف	مستوى الدلالة *p*	حجم التأثير η^2
بين المجموعات	0.354	1	0.354	0.452	0.000	0.7
الخطأ	30.254	45	0.874			

ويلاحظ من الجدول رقم 6 أن مستوى الدلالة للفروق بين متوسطات استجابات أفراد العينة للمحور الثالث من مقياس الفاعلية الذاتية دال احصائياً. وعليه فإنه يتم رفض الفرضية الفرعية الثالثة ويمكن القول بأن الطلاب الذين تلقوا تدريساً في حل المشكلات غير الرياضية ومهارات التفكير الناقد قد اظهروا قدراً أعلى ودالاً إحصائياً من الفاعلية الذاتية المتعلقة بقدرتهم على التنظيم الذاتي أكثر من الطلاب الذين لم يتلقوا تدريساً في حل المشكلات ومهارات التفكير الناقد. كما أن حجم التأثير لهذا الأثر أعلى من المتوسط.

الفرضية 1.4: لا يوجد فروق ذات دلاله إحصائية عند مستوى 0.05 بين الطلاب الذين يتلقون تدريساً في حل المشكلات غير الرياضية ومهارات التفكير الناقد والطلاب الذين لم يتلقوا تدريساً في حل المشكلات ومهارات التفكير الناقد في مستوى الفاعلية الذاتية المتعلقة بقدرتهم على أداء مهام البرمجة البسيطة.

يعرض الجدول رقم 7 المتوسطات والانحرافات المعيارية لاستجابات أفراد المجموعتين التجريبية والضابطة للمحور الرابع والأخير من مقياس الفاعلية الذاتية في البرمجة والذي يركز على قدرة الطالب على أداء المهام البرمجية البسيطة مثل تصميم وتنفيذ برنامج قصير، أو العثور على الأخطاء في برنامج صغير.

جدول 7.

المتوسطات والانحرافات المعيارية لاستجابات أفراد العينة للمحور الرابع من مقياس الفاعلية الذاتية في البرمجة.

المجموعة	التطبيق القبلي			التطبيق البعدي		
	ن	م	إ.م.	ن	م	إ.م.
التجريبية	25	3.258	0.325	25	5.47	0.874
الضابطة	23	3.478	0.698	23	4.12	0.458

ويلاحظ من الجدول رقم 7 تحسن في الفاعلية الذاتية للطلاب من التطبيق القبلي إلى التطبيق البعدي. كما يلاحظ أن أفراد المجموعة التجريبية قد حققوا متوسط درجات أعلى من أفراد المجموعة الضابطة في التطبيق البعدي للأداة. وللتعرف على مستوى دلالة تلك الفروق، فقد تم استخدام اختبار التباين الأحادي.

جدول 8.

اختبار التباين الأحادي لمستوى دلالة الفروق بين متوسطات التطبيق القبلي والبعدي للمحور الرابع من مقياس الفاعلية الذاتية في البرمجة.

المصدر	مجموع المربعات	درجات الحرية *df*	متوسط المربعات	قيمة ف	مستوى الدلالة *p*	حجم التأثير η^2
بين المجموعات	0.187	1	0.187	0.017	0.000	0.7
الخطأ	30.254	45	0.874			

ويلاحظ من الجدول رقم 8 أن مستوى الدلالة للفروق بين متوسطات استجابات أفراد العينة للمحور الرابع من مقياس الفاعلية الذاتية دال إحصائياً. وعليه فإنه يتم رفض الفرضية الفرعية الرابعة ويمكن القول بأن الطلاب الذين تلقوا تدريساً في حل المشكلات غير الرياضية ومهارات التفكير الناقد قد أحرزوا قدراً أعلى ودالاً إحصائياً من الفاعلية الذاتية المتعلقة بقدرتهم على أداء مهام البرمجة البسيطة أكثر من الطلاب الذين لم يتلقوا تدريساً في حل المشكلات ومهارات التفكير الناقد. كما أن حجم التأثير لهذا الأثر أعلى من المتوسط.

وبناء على نتائج اختبار الفرضيات الفرعية 1.1، و 1.2، و1.3، و1.4، فإنه يتم قبول الفرضية الرئيسة الأولى، ويمكن القول بأن الطلاب الذين تلقوا تدريساً في حل المشكلات غير الرياضية ومهارات التفكير الناقد قد حققوا قدراً أعلى ودالاً إحصائياً من الفاعلية الذاتية في مهام برمجة الحاسبات، من الطلاب الذين لم يتلقوا تدريساً في حل المشكلات غير الرياضية ومهارات التفكير الناقد.

الفرضية الثانية: لا يوجد فروق ذات دلاله إحصائية عند مستوى 0.05 بين الطلاب الذين يتلقون تدريساً في حل المشكلات غير الرياضية ومهارات التفكير الناقد والطلاب الذين لم يتلقوا تدريساً في حل المشكلات ومهارات التفكير الناقد في مستوى الفاعلية الذاتية في مستوى **الأداء الأكاديمي.**

يعرض الجدول رقم 9 المتوسطات والانحرافات المعيارية للدرجات لنهائية لأفراد المجموعتين التجريبية والضابطة والتي تشمل درجاتهم في الاختبار النهائي، والاختبارات القصيرة، والواجبات البرمجية.

جدول 9.

المتوسطات والانحرافات المعيارية لاستجابات أفراد العينة للمحور الرابع من مقياس الفاعلية الذاتية في البرمجة.

المجموعة	ن	الواجبات (20 درجة)	الاختبارات القصيرة (5 درجات)	الاختبار النصفي (35 درجة)	الاختبار النهائي (40 درجة)	المجموع (100)	الانحراف المعياري
التجريبية	25	17.5	4.75	29	35	86.25	2.587
الضابطة	23	12.75	2.75	22.5	25	63	3.024

ويلاحظ من الجدول رقم 9 أن أفراد المجموعة التجريبية قد حققوا متوسط مجموع درجات أعلى من أفراد المجموعة الضابطة. وللتعرف على مستوى دلالة تلك الفروق، فقد تم استخدام اختبار التباين الأحادي.

جدول 10.

اختبار التباين الأحادي لمستوى دلالة الفروق بين متوسطات التطبيق القبلي والبعدي للمحور الرابع من مقياس الفاعلية الذاتية في البرمجة.

المصدر	مجموع المربعات	درجات الحرية *df*	متوسط المربعات	قيمة ف	مستوى الدلالة *p*	حجم التأثير η^2
بين المجموعات	2548	1	2548	19.347	0.001	0.9
الخطأ	4029	45	14.025			

ويلاحظ من الجدول رقم 10 أن مستوى الدلالة للفروق بين متوسطات الدرجات النهائية لأفراد العينة دال احصائياً. وعليه فإنه يتم رفض الفرضية الثانية ويمكن القول بأن الطلاب الذين تلقوا تدريساً في حل المشكلات غير الرياضية ومهارات التفكير الناقد قد حققوا قدراً أعلى ودال إحصائياً من **الأداء الأكاديمي** مقارنة بالطلاب الذين لم يتلقوا تدريساً في حل المشكلات غير الرياضية ومهارات التفكير الناقد. كما أن حجم التأثير لهذا الأثر أعلى من المتوسط.

مناقشة النتائج

جاءت نتائج هذه الدراسة متفقة مع ما توصلت له الدراسات السابقة من حيث الأثر الإيجابي لتدريس مهارات حل المشكلات على الفاعلية الذاتية والأداء الأكاديمي للطلاب. فقد أظهرت نتائج هذه الدراسة أن تدريس الطلاب مهارات حل المشكلات والتفكير الناقد غير الرياضية قد أديا إلى تحسن ملحوظ في أدائهم الأكاديمي في مقرر برمجة الحاسبات وفاعليتهم الذاتية في مجال البرمجة، ويدعم ذلك نتائج الدراسات التي توصل لها كل من (2009) Arshad و Baker et al. (2007) و Beckwith et al. (2006) . ويمكن تفسير هذه النتائج بأنه عندما يتم الطلاب اختباراتهم القصيرة وواجباتهم واختباراتهم بنجاح فإن ذلك يؤدي إلى زيادة الفاعلية الذاتية لديهم. كما دلت الدراسات السابقة على أن دمج مفردات في مهارات حل المشكلات مع مفردات المقرر الرئيسية يساعد على النجاح في المقررات المستقبلية (Beaubouef & McDowell, 208; DeClue, 2008; Gold & Rimmer, 2000).

وقد تعزى هذه النتائج كذلك إلى أن تدريس برمجة الحاسبات بطريقة حل المشكلات وفرت مناخاً مناسباً للطلاب، وبيئة تعليمية غنية بالمعززات الداخلية. كما أنها نقلت الطالب من الدور السلبي إلى لدور الإيجابي عندما كان الطالب يبحث عن المعرفة، ويكتشفها بنفسه، بدلاً من أن تعطى له جاهزة، فتهيئه المواقف التعليمية المتنوعة كان له الدور البارز في صقل إمكانياتهم وقدراتهم مما زاد من فاعليتهم الذاتية، ومتابعة ما تتضمنه تلك الطريقة من أنشطة تعليمية، وذلك بدوره يساعد على فهم تلك المعرفة من جهة، واحتفاظهم بها من جهة أخرى. كما أن طرح موضوعات دروس البرمجة على شك مشكلات مرتبطة ببيئة الطالب، أو عمله المستقبلي قد ساعد على زيادة اهتمام الطلاب بالتعلم، وتضييق الفجوة بين الموقف التعليمي والموقف الحياتي مما زاد في اقبالهم على المحتوى التعليمي، وهذا بدوره ساعد على رفع مستوى التحصيل الدراسي والفاعلية الذاتية. وهذا ما أكدته ياسين (2011) حيث أشارت إلى أن طريقة حل المشكلات توفر للمتعلمين الإثارة والدافعية للتعلم.

كما يمكن أن تعود النتائج الإيجابية التي توصلت إليها هذه الدراسة إلى أن طريقة حل المشكلات أتاحت للطلاب فرصاً متعددة للتفكير والبحث عن حلول مختلفة للمشكلات البرمجية، وأدى ذلك إلى تنوع الأفكار وإثرائها لدى الطلاب. كما ساهمت طريقة حل المشكلات في توليد الأفكار والخبرات الجديدة، فسلك الطلاب المنهج العلمي لبحث عن الحلول العملية. ويؤكد ذلك عطا الله (2002) بقوله إنه من خلال طريقة حل المشكلات يسلك الطلاب سلوك العلماء في البحث والتقصي، مما ينتج عنه التفكير السليم المبدع.

وقد وجد كل من ناوتا وابيرسون (Nauta & Epperson, 2003) أنه يمكن التأثير على الفاعلية الذاتية للطالب بشكل ايجابي في المرحلة الثانوية والجامعية، وخاصة إن كان تحصيل

الطالب ودافعيته تدعمان بقاءه في التخصص. ويرى مايال (Mayall, 2008) بأنه بمكن التأثير بشكل ايجابي على الفاعلية الذاتية لطلاب الجامعة، وقد أثبتت نتائج هذه الدراسة بأنه يمكن التأثير ايجابياً على الفاعلية الذاتية في مجال الحاسب عن طريق التدريس المباشر لمهرات حل المشكلات. وقد أكد علي (Ali, 2005) على أهمية تدريس حل المشكلات والتفكير الناقد كجزء من مقررات البرمجة الأولية في تخصصات الحاسب

كما أظهرت نتائج هذه الدراسة أن التذكير المستمر خلال الفصل الدراسي بأهمية استخدام مهارات حل المشكلات ربما أدى إلى تحسن أداء الطلاب أكاديمياً. فقد قام مدرسو البرمجة بتذكير الطلاب بشكل متواصل خلال الفص الدراسي بأهمية استخدام مهارات حل المشكلات التي تعلموها في بداية الفصل الدراسي. وقد أوصى ماديقان وآخرون (Madigan et al., 2007) بأن يجد معلمو علوم الحاسب الآلي طرقاً لبناء الثقة تدريجياً لدى الطلاب من خلال مقررات علوم الحاسب، ولكنهم لا يقدمون اقتراحات حول الاستراتيجيات الفعالة التي يمكن اتباعها لبناء تلك الثقة. وقد أظهرت نتائج هذه الدراسة بأن التذكير المستمر باستخدام مهارات حل المشكلات يعتبر استراتيجية مناسبة لتعزيز الفاعلية الذاتية والتحصيل.

بالإضافة إلى ما سبق، فقد تم استخدام المجموعات في تدريس مهارات حل المشكلات كما أوصى بذلك علي وجبسون (Ali, 2005; Gibson & O'kelly, 2005). فمن الضروري التركيز ليس فقط على مهارات حل المشكلات، وإنما مهارات حل المشكلات داخل المجموعات الطلابية عند تدريس مقررات البرمجة، إذ يعتبر ذلك محاكاة لبيئة العمل التي سيعمل بها الطالب بعد تخرجه. ويتفق مع ما سبق كل من فالكنر وبالمر (Falkner & Palmer, 2009). إذ أوصوا بتضمين محاضرات في حل المشكلات، حيث يمارس الطلاب نشاطات حل المشكلات، يتبعها نشاطات تعاونية في حل المشكلات. ويرى فالكنر وبالمر أن جانب العمل الجامعي في حل المشكلات مهم للتطبيق الفعلي لما يتعلمه الطالب في العالم الخارجي أثناء عمله، وأنه لا يمكن إغفال هذال الجانب.

وجملة القول، فإنه يتضح من نتائج هذه الدراسة الأثر الإيجابي لاستخدام أسلوب حل المشكلات في تدريس برمجة الحاسبات، وعلى الفاعلية الذاتية للطلاب، وتفوقه في ذلك على الطريقة التقليدية في التدريس. وقد يعزى السبب في ذلك إلى كون تعلم حل المشكلات هو تعلم استراتيجيات في التفكير قابلة للتطبيق والانتقال إلى موافق أخرى. هذا فضلاً عن أن العمل الجماعي الذي يتضمنه أسلوب حل المشكلات يتيح الفرصة أمام الطلبة لتبادل الأفكار والاقتراحات والخبرات فيما بينهم خلال أدائهم لنشاطات التعلم، وتشارك ما يجول في أذهانهم من أفكار وآراء، والتعبير عن عمليات التفكير التي يستخدمونها أثناء أدائهم لنشاطات البرمجة المختلفة، سواء كان ذلك التعبير كتابة أم لفظاً. ومن ثم حصيلة التفاعل والنقاشات بين الطلبة

خروج كل طالب بجملة من الأفكار والآراء والاستراتيجيات التفكيرية التي تم تبادلها مع الطلبة، مما سيؤدي إلى تنمية وتعزيز قدراتهم في برمجة الحاسبات.

التوصيات

بناء على النتائج التي توصلت إليها هذه الدراسة، فإنه يمكن التوصية بما يلي:

1. ضرورة تضمين مقررات المدخل إلى البرمجة مفردات يمكن للطلاب دراستها ذاتياً حول حل المشكلات غير الرياضية، وقد وجد في هذه الدراسة أن تدريس حل المشكلات غير الرياضية عبر نظام إدارة التعلم قد أثبت فعاليته.
2. ضرورة استخدام معلمي الحاسب الآلي لأسلوب حل المشكلات غير الرياضية في تدريسهم للغات البرمجة لطلاب المرحلة الجامعية أو الثانوية، لتسهل عليهم عملية التعلم.
3. الحاجة إلى تدريب الطلاب على استخدام استراتيجيات حل المشكلات غير الرياضية من خلال ما يقدم لهم من أنشطة، وبرامج، ومقررات دراسية.
4. الحاجة إلى تدريب أعضاء هيئة التدريس في تخصصات الحوسبة، ومعلمي الحاسب في المرحلة الثانوية لتدريبهم على استخدام طريقة حل المشكلات في تخطيط وتنفيذ الدروس.
5. الحاجة إلى إعادة النظر في الأساليب الحالية في تدريس برمجة الحاسبات، أخذا في الاعتبار تخوف الطلاب من هذه المقررات، وعدم وجود خبرات سابقة في البرمجة تساعدهم على تكوين تصور حول مدى قدرتهم على التفوق في تلك المقررات.
6. الحاجة إلى تضمين التعلم الجماعي إلى استراتيجيات التعلم في مقررات برمجة الحاسبات، فقد تعلم الطلاب في هذه الدراسة مهارات حل المشكلات جماعياً مع أقرانهم.
7. الحاجة إلى إجراء دراسة مماثلة على عينة من الطالبات الإناث، فقد أشارت بعض الدراسات إلى وجود اختلاف في الفاعلية الذاتية والتحصيل لأكاديمي بين الإناث والذكور في مقررات برمجة الحاسبات.
8. الحاجة إلى إجراء دراسة مماثلة في مقررات برمجة قواعد البيانات والذكاء الاصطناعي وشبكات الحاسب.
9. الحاجة إلى إجراء المزيد من الدراسات التي تتناول أثر أسلوب حل المشكلات على متغيرات أخرى كالاتجاهات، وذلك في مادة الحاسب الآلي أو غيرها من المواد الدراسية، ولصفوف مراحل دراسية مختلفة.

10. الحاجة إلى إجراء دراسات تجريبية مماثلة للتعرف على فاعلية أسلوب حل المشكلات غير الرياضية في زيادة التحصيل الأكاديمي والفاعلية الذاتية عند تدريس موضوعات أخرى في المواد العلمية كالكيمياء والفيزياء.

المراجع

أحمد، يمنى سمير عبدالوهاب . (2011). برنامج حاسوبي قائم على حل المشكلات و أثره في تنمية الاستعداد للقراءة لدى أطفال الروضة . *مجلة القراءة والمعرفة –مصر ,ع 113,* , ص ص.107 – 123.

جامع ، حسن حسيني . (2012). فعالية التدريس الخصوصي بالكمبيوتر في تنمية مهارات حل المشكلات البرمجية لدى طلاب كلية التربية النوعية . *المجلة العربية للتربية العلمية – اليمن , مج 1, ع 1,* , ص ص.98 – 132.

حسن ، رسول عباس . (2012). أثر استخدام طريقة حل المشكلات في تدريس مادة عناصر الفن . *مجلة كلية التربية الأساسية* – كلية التربية الأساسية – الجامعة المستنصرية – العراق,ع 75 , ,ص ص.521 – 550 .

شريف ، محمد أنور عبدالرحمن أحمد . (2011). فعالية استراتيجية التعلم المتمركز حول المشكلة في تنمية المهارات العملية والقدرة على حل المشكلات والاتجاه نحو العمل التعاوني في مادة تكنولوجيا الكهرباء لدى طلاب الصف الثاني الثانوي الصناعي .*مجلة كلية التربية بالمنصورة* –مصر , ع 75, ج 1 , ,ص ص.338 – 369 .

نجم ، خميس موسى . (2011). أثر استخدام أسلوب حل المشكلات في تدريس الرياضيات في تنمية التفكير الناقد لدى طلبة الصف التاسع الأساسي . *المجلة التربوية –الكويت , مج 25, ع 98,* , ص ص.201 – 230. دار المسيرة، عمان، الأردن.

عطا الله، ميشيل كامل. (2002). *طرق وأساليب تدريس العلوم*. دار المسيرة، عمان، الأردن.

ياسين، ثناء محمد أحمد . (2013). فاعلية طريقة حل المشكلات في العلوم التطبيقية على التحصيل الدراسي و تنمية مهارات التفكير الابداعي لدى طالبات الصف الثاني المتوسط بمدينة مكة المكرمة . *مجلة جامعة أم القرى للعلوم التربوية والنفسية – السعودية , مج 5, ع 1* , ,ص ص.63 – 142 .

Adya, M.P. (2008). *Work alienation among IT workers: A cross-cultural gender comparison*. Proceedings of the ACM SIGMIS-CPR 2008 Conference, 66 – 69 .

Al-Bow, M., Austin, D., Edgington, J., Fajardo, R., Fishburn, J., Lara, C., et al .(2008) *Proceedings of the 2008 ACM SIGGRAPH symposium on Video games*, 55 – 59 .

Ali, S. (2005). Effective teaching pedagogies for undergraduate computer science *Mathematics and Computer Science Education*, 39(3), 243 – 257 .

Allan, V. & Kolesar, M. (1997). Teaching computer science: A problem solving approach that works. *ACM SIGCUE Outlook*, 25(1), 2 – 10.

Baker, D., Krause, S., Yasar, S., Roberts,C., & Robinson-Kurpius, S. (2007). An intervention to address gender issues in a course on design, engineering, and technology for science educators. *Journal of Engineering Education*, 96(3), 213 – 226.

Ballou, D.J. & Huguenard, B.R. (2008). The impact of students' perceived computer experience on behavior and performance in an introductory information systems course. *Journal of Information Systems Education*, 19(1), 87 – 97.

Bandura, A. (1994). Self-efficacy. InV.S. Ramachaudran (Ed.), Encyclopedia of human behavior (Vol. 4, pp. 71 – 81). New York: Academic Press.

Barker, L., McDowell, C. & Kalahar, K. (2009). *Exploring factors that influence computer science introductory course students to persist*. Proceedings of the ACM SIGCSE Technical Symposium on Computer Science Education '09, 153 – 157.

Beaubouef, T. & McDowell, P. (2008). Computer science: Student myths and misconceptions. *Journal of Computing Sciences in Colleges*, 23(6), 43 – 48.

Beaubouef, T., Lucas, R., & Howatt, J. (2001). *The UNLOCK system: Enhancing problem solving skills in CS-1 students.* ACM SIGCSE Bulletin, 33(2), 43 – 46.

Beckwith, L., Kissinger, C., Burnett, M., Wiedenbeck, S., Lawrance, J., Blackwell, A., et al. (2006). *Tinkering and gender in end-user programmers' debugging*. Proceedings of the ACM CHI 2006 Conference, 231 – 240.

Bennedsen, J., & Caspersen, M. E. (2008). Optimists have more fun, but do they learn better? On the influence of emotional and social factors on learning introductory computer science. *Computer Science Education*, 18(1), 1 – 16.

Biggers, M., Brauer, A., & Yilmaz, T. (2008). *Student perceptions of computer science: A retention study comparing graduating seniors vs. CS leavers*. Proceedings of the ACM SIGCSE technical Symposium on Computer Science Education '08, 402 – 406.

Cady, D. & Terrell, S. (2007). The effect of the integration of computing technology in a science curriculum on female students' self-efficacy attitudes. *Journal of Educational Technology Systems*, 36(3), 277 – 286.

Cassidy, S. & Eachus, P. (2002). Developing the computer user self-efficacy (CUSE) scale: Investigating the relationship between computer self-efficacy, gender and experience with computers. *Journal of Educational Computing Research*, 26(2), 133 – 153.

Cho, M. (1995). *Turning Point for Korean Computer Educators: Introducing LogoWriter as a Thinking Tool*. Proceedings of the National Educational Computing Conference: Emerging Technologies, Lifelong Learning, NECC '95, 237 – 242.

Cohoon, J., Wu, Z., & Luo, L. (2008). *Will they stay or will they go*? Proceedings of the ACM SIGCSE technical symposium on Computer Science Education '08,397 – 401.

Colley, A., Henry, O., Holmes, S., & James, L. (1996). Perceptions of ability to program or to use a word processor. *Computers in Human Behavior*, 12(3), 329 - 337.

Courte, J. & Bishop-Clarke, C. (2009). *Do students differentiate between computing disciplines?* Proceedings of the 40th ACM Technical Symposium on Computer Science Education, 29 – 33.

DeClue, T. (2008). Computer science in Kindergarten? Of course! The ABCs of the K-12 CSTA model curriculum in computer science. *Journal of Computing Sciences in Colleges*, 23(4), 257 – 262.

Dijkstra, E. W. (n.d.) *An effort towards structuring of programmed processes*.[EWD 198.] Retrieved February 17, 2013, from the University of Texas E. W. Dijkstra Archive: http://www.cs.utexas.edu/users/EWD/ewd01xx/EWD198.PDF

Downey, J. (2006). *Measuring general computing self-efficacy: The surprising comparison of three instruments in predicting performance, attitudes, and usage*. Proceedings of the 39th Hawaii International Conference on System Sciences.

Eastman, E. (2003). Fact-based problem identification precedes problem solving. *Journal of Computing Sciences in Colleges*, 19(2), 18 – 29.

Edmondson, C. (2008). *Real women don't write programs*. ACM SIGCSE Bulletin, 40(2), 112 – 114.

Faulkner, K. & Palmer, E. (2009). *Developing authentic problem solving skills in introductory computer science courses*. Proceedings of the ACM SIGCSE Technical Symposium on Computer Science Education '09, 4 – 8.

Gibson, J. & O'Kelly, J. (2005). *Software engineering as a model of understanding for learning and problem solving*. Proceedings of

the 2005 international workshop on computing education research ICER '05, 87 – 97.

Goold, A. & Rimmer, R. (2000). *Indicators of performance in first-year computing*. Proceedings of the IEEE Computer Science Conference 2000. 23rd Australasian, 74 – 80.

Gries, D. (1974). *What should we teach in an introductory programming course*? In Proceedings of the Fourth SIGCSE Technical Symposium on Computer Science EducationSIGCSE '74 (pp. 81-89). New York: Association for Computing Machinery.

Hanks, B. & Brandt, M. (2009). *Successful and unsuccessful problem solving approaches of novice programmers*. Proceedings of the ACM SIGCSE Technical Symposium on Computer Science Education ’09, 24 – 28.

Hart, M., Early, J., & Brylow, D. (2008). *A novel approach to K-12 CS education: Linking mathematics and computer science*. Proceedings of the ACM SIGCSE technical symposium on Computer Science Education ’08, 286 – 290.

Hazzan, O., Gal-Ezer, J., & Blum, L. (2008). *A model for high school computer science education: The four key elements that make it!* Proceedings of the ACM SIGCSE technical symposium on Computer Science Education ’08, 281 – 285.

Jin, W. (2008). *Pre-programming analysis tutors help students learn basic programming concepts*. Proceedings of the ACM SIGCSE technical symposium on Computer Science Education ‘08, 276 – 280.

Joshi, K.D. & Schmidt, N.L. (2006). *Is the information systems profession gendered? Characterization of IS professional and IS careers*. The DATA BASE for Advances in Information Systems, 37(4), 26 – 41.

Klappholz, D. (2009). Organizing and delivering “real projects for real clients” courses. *Journal of Computing Sciences in Colleges*, 24(3), 106 – 108.

Klawe, M. & Shneiderman, B. (2005). Crisis and opportunity in computer science. Communications of the ACM, 48(11), 27 – 28.

Kranch, D. A. (2010). *A study of three instructional sequences for developing computer programming experties in novice learners*. Unpublished doctoral Dissertation. Capell University.

Kumar, A.N. (2008). *The effect of using problem-solving software tutors on the self-confidence of female students*. Proceedings of the ACM SIGCSE technical symposium on Computer Science Education ’08, 523 – 527.

Kurland. D. M., Pea. R. D., Clement. C., & Mawby, R. (1989). A study of the development of programming ability and thinking skills in high school students. In E. Soloway & J.C. Spohrer (Eds.), *Studying the novice programmer* (pp. 83-112). Hillsdale. NJ: Lawrence Erlbaum.

Lai, Y., & Wong, T. (2007). *Infusing problem-solving skills into computer lessons*. ACM SIGCSE Bulletin, 39(4), 84 – 86.

Lemire, D. (2002). Math problem solving and mental discipline – The myth of transferability. Journal of College Reading and Learning, 32(2), 229 – 238.

Linn. M. C., & Dalbey, J. (1989). Cognitive consequences of programming instruction. In E. Soloway & J.C. Spohrer (Eds.), Studying the novice programmer (pp. 57-81). Hillsdale. NJ: Lawrence Erlbaum.

Liu, M., Hsieh, P., Cho, Y., & Schallert, D. (2006). Middle school students' self-efficacy, attitudes, and achievement in a computer-enhanced problem-based learning environment. Journal of Interactive Learning Research, 17(3), 225 – 242.

Madigan, E.M., Goodfellow, M., & Stone, J.A. (2007). Gender, perceptions, and reality: Technological literacy among first-year students. Proceedings of the ACM SIGCSE Technical Symposium on Computer Science Education '07, 410 – 414.

Mayall, H. J. (2008). Difference in gender based technology self-efficacy across academic levels. International Journal of Instructional Media, 35(2), 145 – 155.

McGettrick, A., Boyle, R., Ibbett, R., Lloyd, J., Lovegrove, G., & Mander, K. (2005). Grand challenges in computing: Education – A summary. The Computer Journal, 48(1), 42-48.

Moskal, B., Lurie, D., & Cooper, S. (2004). Evaluating the effectiveness of a new instructional approach. Proceedings of the 35th SIGCSE Technical Symposium on Computer Science Education, 75 – 79.

Mullins, P., Whitfield, D., & Conlon, M. (2008). Using Alice 2.0 as a first language. *Journal of Computing Sciences in Colleges*, 24(3), 136 – 143.

Nauta, M. M, & Epperson, D. L. (2003). A longitudinal examination of the social-cognitive model applied to high school girls' choices of nontraditional college majors and aspirations. *Journal of Counseling Psychology*, 50(4), 448 – 457.

Norris, C., Barry, F., Fenwick, J., Reid, K., & Rountree, J. (2008). ClockIt: *Collecting quantitative data on how beginning software developers really work*. Proceedings of the 2008 ITiCSE Conference, 37 – 41.

Olsen, A. (2005). Using pseudocodeto teach problem solving. *Journal of Computing Sciences in Colleges*, 21(2), 231 – 236.

Palumbo, D.B. (1990). Programming language / problem-solving research: A review of relevant issues. *Review of Educational Research*, 60(1), 65 – 89.

Pintrich, P. & Schunk, D., (1996). *Motivation in education: Theory, research, & applications*. Englewood Cliffs, NJ: Prentice Hall.

Pulimood, S.M. & Wolz, U. (2008*). Problem solving in community: A necessary shift in CS pedagogy.* Proceedings of the ACM SIGCSE Technical Symposium on Computer Science Education '08, 210 – 214.

Putnam, R T., Sleeman, D., Baxter, J. A., & Kuspa, L. K. (1989). *A summary of misconceptions of high school basic programmers*. In E. Soloway & J.C. Spohrer (Eds.), studying the novice programmer (pp. 301-315). Hillsdale. NJ: Lawrence Erlbaum.

Quade, A. (2003). Development and validation of a computer science self-efficacy scale for CS0 courses and the group analysis of CS0 student self-efficacy. Proceedings of the International Conference on Information Technology: Computers and Communications (ITCC'03), 60 – 64.

Rafieymehr, A. (2008). Kids in computing (K.I.C.): *Is there a solution to solve the computer science enrollment problem?* ACM SIGCSE Bulletin, 40(2), 107 – 111.

Ramalingam, V. & Wiedenbeck, S. (1998). Development and validation of scores on a computer programming self-efficacy scale and group analysis of novice programmer self-efficacy. *Journal of Educational Computing Research*, 19(4), 367 – 381.

Ramalingam, V. & Wiedenbeck, S. (1998). Development and validation of scores on a computer programming self-efficacy scale and group analysis of novice programmer self-efficacy. *Journal of Educational Computing Research*, 19(4), 367 – 381.

Rao, M.R.K. (2006). Storytelling and puzzlesin a software engineering course. Proceedings of the ACM SIGCSE Technical Symposium on Computer Science Education '06, 418 – 422.

Robins, A., Rountree, J., & Rountree, N. (2003). Learning and teaching programming: A review and discussion. *Computer Science Education*, 13(2), 137-172.

Sands, M., Moukhine, N., & Blank, G. (2008). Widening the pipeline of K-12 students with Flash. *Journal of Computing Sciences in Colleges*, 23(5), 52 – 57.

Scragg, G. & Smith, J. (1998). A study of barriers to women in undergraduate computer science. Proceedings of the ACM SIGCSE technical Symposium on Computer Science Education '98, 82 – 86.

Sivilotti, P.A.G. & Laugel, S.A. (2008). *Scratching the surface of advanced topics in software engineering: A workshop module*

for middle school students. Proceedings of the ACM SIGCSE technical symposium on Computer Science Education ’08, 291 – 295.

Soloway, E., Erlich. K., Bonar, J., & Greenspan, J. (1982). What do novices know about programming? In A. Badre & B. Shneiderman (Eds.), Directions in human-computer interaction (pp. 27-54). Norwood, NJ: Ablex.

Tang, M., Pan, W., & Newmeyer, M. D. (2008). Factors influencing high school students’ career aspirations. *Professional School Counseling*, 11(5), 285 – 295.

Van Sickle, E. (2008). *Refilling the IT pipeline and using storage technologies as a specialization.* Proceedings of the ACM SIGMIS-CPR 2008 Conference, 112 – 118.

Whimbley, A. & Lochhead, J. (1999). *Problem solving & comprehension*. 6th Edition. Mahwah, N.J.: Lawrence Erlbaum Associates.

Wilson, B. (2008). Improving comfort level of females in the first computer-programming course: Suggestions for CS faculty. *Journal of Computing Sciences in Colleges*, 23(4), 28 – 34.

Winslow, P. (1996). Why some material is difficult to learn. *Cognition and Instruction*, 12(3), 185-233.

The effect of non-mathematical problem solving instruction on students' computer programming self-efficacy and academic achievement in an introduction to programming course

Riyadh Alhassan Ph. D.
King Saud University- Saudi Arabia

Abstract: This study aimed to investigate the effect of teaching non-mathematical problem solving skills on self-efficiency and academic performance of students in introduction to programming. This study has been limited to a sample of male students studying in computer programming (1), and conducted during the first semester of the academic year 2013. This study followed a quasi-experimental design containing a pre- and post-test and two groups (experimental and control). The study was conducted on two sections of introduction to computer programming. Students have completed the self-efficacy scale in computer programming before the beginning of the course. And the control group was taught using teaching method in which the instructors usually teach, without special training about problem-solving. The experimental group received additional instruction in problem solving at the beginning of the semester with a focus on critical thinking skills necessary to solve programming problems. In the middle of the semester, and at the end students competed a measure of self-efficacy in computer programming again. The results of the study have shown that the experimental Group scored higher than the control group in the self-efficacy scale in computer programming> Moreover, the experimental group obtained higher scores in academic achievement, and the differences between the experimental and control groups were statistically significant.

Keywords: self-efficacy in computer programming, non-mathematical problem solving, computer programming